

Ecole Nationale Supérieure d'Architecture Paris la Villette  
Ecole Doctorale Ville Transports et Territoires, PRES Paris Est

**Thèse**

pour obtenir le grade de  
Docteur de l'Université Paris Est  
Section : Architecture

**Modélisation paramétrique  
en conception architecturale :  
Caractérisation des opérations cognitives de  
conception pour une pédagogie**

Présentée par :

Aurélie de Boissieu  
Laboratoire MAP-maacc  
(juin 2013)

Sous la direction de :

François Guéna (directeur de thèse), Professeur titulaire des écoles d'architecture, lab. MAP-maacc  
Caroline Lecourtois (co-directrice de thèse), Maître assistant titulaire des écoles d'architecture, lab. MAP-maacc

Jury :

Pierre Leclercq (rapporteur), Professeur à l'Université de Liège, lab. LUCID Group  
Gilles Halin (rapporteur), Professeur à l'Université de Lorraine, lab. MAP-crai  
Philippe Boudon, Professeur titulaire des écoles d'architecture  
Antoine Picon, Professeur à Harvard University Graduate School of Design, lab. GSD, LATTIS  
Benachir Medjdoub, Professeur à Nottingham Trent University



## Remerciements

La réalisation de cette thèse doit beaucoup, et sur de nombreux plans, à mes directeur et co-directeur de thèse, à mes collègues, comme à ma famille et à mes amis. Il est difficile de rendre compte de tout ce que ce travail doit, mais je tiens à remercier ici ceux sans qui je n'aurais pu réaliser ce travail. Je leur en suis d'autant plus reconnaissante que ce travail vise à être non pas un aboutissement mais un prémisses vers ce à quoi j'aspire : une ouverture à la pratique de la recherche.

Ma gratitude va en premier lieu à François Guéna et Caroline Lecourtois. Je les remercie tous deux pour leur encadrement attentif. Je tiens également à remercier Anne-Sophie Delaveau et Samia Ben Rajeb, collègues et amies.

Je remercie messieurs les membres de mon jury : Pierre Leclercq, Gilles Halin, Philippe Boudon, Antoine Picon et Benachir Medjdoub de m'avoir fait l'honneur de participer à l'évaluation et à la validation de ma recherche. Grâce à eux, ce travail de recherche va s'inscrire dans la *disputatio* scientifique. Parmi eux, je remercie tout particulièrement mes rapporteurs Gilles Halin et Pierre Leclercq.

Je tiens également à remercier les différents chercheurs qui, par leur présence bienveillante et leurs discussions m'ont aidé à murir ma recherche. Ma gratitude va tout particulièrement à Philippe Boudon, Bernard Duprat, Thierry Ciblac et Catherine Deshayes. Elle va également à Louis-Paul Untersteller, Yves Mineur et Marc Daniel pour leurs aides sur les mécanismes informatiques de la modélisation paramétrique. Je dois également beaucoup à Elise Macaire, Emmanuel Amougou Mbala, Isabelle Grudet, Mickael Fenker et Christophe Camus. Je souhaite également remercier Rémy Mouterde, François Tran, Hervé Lequay, Philippe Marin et Gilles Desèvedavy pour m'avoir initié à la recherche lors de mon Master à l'ENSA de Lyon. Je souhaite également remercier tous les acteurs de DNArchi (comité de rédaction, auteurs, lecteurs et relecteurs) de permettre la réalisation de ce beau projet collectif.

Je souhaite également remercier tous ceux qui se sont pliés à l'exercice de l'entretien et grâce à qui cette recherche a pu être menée. Ma gratitude va à : Emilie Hergott (RFR), Lorenzo Ponzo (TESS), Nicolas Leduc (RFR), Greighton Willis (Studios Architecture), Bernard Cache (Objectile), Andrew Witt (GT), Paul Ehrett (GT), Hugh Whitehead (SMG, F+P), Marion Ott (THP), Miriam Dalligna (F+P), Elodie Le Roy (THP), Justin Diles (THP), Christian Girard (THP), Philippe Morel (THP), Adam Orliensky (X-Over), Philip Hornung (X-Over), Siim Tuksam (X-Over), Milovann Yanatchkov (RVBA), Angelos Chronis (ARD, F+P), Francis Aish (ARD, F+P), Sam Joyce (ARD, F+P), Gaétan Koehler (HDA), Alexandre Pachiaudi (RPBW), Francesco Cingolan (RDA), Rick Smith (GP), Toshihiro Kubota (AJN), Hugh Dutton (HDA), Pierluigi Bucci (HDA), Gilles Desèvedavy (ReSie), Valerio Bonora (Decode) ;

Je remercie pour m'avoir accompagné dans mes premières expériences d'enseignements : Thierry Ciblac, François Guéna (ENSA de Paris la Villette), Suzanne Monnot, Alena Kubova (ENSA de Lyon), Nadir Tazdait, Francesco Cingolani (ENSA de Marne), Claire Petetin, Julien Véga, Pierre Antoine, Antoine Fourneau et Vincent Roudaut (ENSA de Versailles).

Je souhaite également remercier pour leur aide précieuse de relecture : Caroline Lecourtois, François Guéna, Samia Ben Rajeb, Frédérique Lapillonne, Loise Lenne et Pierre Sauvage.

De façon plus personnelle, je souhaite remercier pour leur amitié dans cette aventure psychologique qu'est la thèse : Joaquim, Sébastien, Milovann, Taichi, Samia et Anne-Sophie pour nos rendez-vous « digikids », le comité d'administration des Cafés de l'Après Thèse de 2010 à 2012 : Anna, Hélène, Francesca, Anastasia et Pauline ; ainsi que Loise Lenne, Marion Roussel et Nazila Hannachi.

Je souhaiterais remercier tout particulièrement ma famille et mes amis pour leur soutien et leur amour. Je ne peux les nommer tous, mais je remercie tout particulièrement mes parents mon frère et ma sœur. Je remercie mes amis chers pour leur amitié inconditionnelle : Marine, Vincent, Cécile, Clémentine, Géraldine, Blandine, Marie. Je remercie tout particulièrement Pierre et Guillaume pour leurs soutiens.

Je dédie ce travail à Raoul qui n'aura pas survécu à la thèse et qui, dans le silence de son bocal, aura été au plus près de cette expérience.



## Résumé

Cette recherche porte sur les usages des logiciels dit de « modélisation paramétrique » en conception architecturale. Plus spécifiquement, sont interrogées les activités cognitives de conception impliquées lors de ces usages. L'architecturologie est le support scientifique de cette recherche. Les concepts de l'architecturologie permettent d'interroger les activités cognitives de conception en termes d'opérations d'attribution de mesures.

Les résultats de cette thèse montrent que les principales opérations cognitives de conception de modèles paramétriques sont :

- des *opérations élémentaires de la conception* telles qu'interrogées par l'architecturologie: le *découpage*, le *dimensionnement* et la *référenciation* ;

- des opérations « pragmatiques », c'est-à-dire participant de la conception mais ne relevant pas directement d'une opération d'attribution de mesures, ces opérations sont : la *mise en commun*, *l'interprétation*, *l'appropriation d'outil* et l'opération de *traduction en géométrie paramétrique* ;

des opérations de logiques relevant de l'induction ou de la vérification.

Ces opérations nécessitent et participent à la construction d'un *espace des savoirs de la modélisation paramétrique*. Cet espace des savoirs ainsi que les opérations pragmatiques observées permettent une intrication par l'échange et le partage de *pertinences* et *d'espaces de référence* entre la conception de modèles paramétriques et la conception architecturale.

Les analyses menées pointent les difficultés rencontrées dans ces activités de conception.

La caractérisation des opérations de conception propres à l'usage de la modélisation paramétrique en conception architecturale a permis d'interroger la méthode des patterns proposée par R. Woodbury. Ces patterns sont proposés comme une aide à la conception de modèles paramétriques en conception architecturale. L'analyse des usages de ces patterns montre que si ceux-ci sont potentiellement une aide puissante, leur généricité les rend difficiles d'accès. Cette méthode peut être complétée par le développement d'une bibliothèque de patterns appliqués (appelés samples) permettant à l'étudiant un apprentissage par l'exemple et par induction. Un support d'apprentissage s'appuyant sur ces résultats a été développé et testé lors de la thèse sur la plateforme [www.parametric-ressources.com](http://www.parametric-ressources.com).

Les connaissances construites dans cette thèse sur l'usage de la modélisation paramétrique en conception architecturale a permis de proposer une pédagogie visant à accompagner l'apprentissage des activités cognitives sollicitées. Des objectifs pédagogiques ont été définis par rapport aux opérations cognitives que l'étudiant doit pouvoir mettre en œuvre. Une pédagogie par le projet ainsi qu'une pédagogie par travaux dirigés sont proposées en vue de répondre à ces objectifs.

**Mots clefs : Architecturologie, conception architecturale, modélisation paramétrique, pédagogie**



## Abstract

This research focuses on the uses of parametric modeling in architectural design. The object of this research is to identify cognitive operations involved in these uses. The characterisation of these mental operations is then a support to develop pedagogy and didactics tools.

Analyses of design practices in professional contexts show that parametric modeling is linked to various computer assisted tasks: complex form finding and representation, evaluation, optimization, fabrication, communication, collaboration etc. We observed in our analyses that most often parametric modeling actors are not the one who design projects. Parametric modeling requires expert skills and knowledge that most architects have not yet mastered. How architects could be assisted for parametric modeling during the design process and how they could be trained are thus crucial questions.

In order to address these questions we searched to identify the characteristics of the cognitive operations implied in the uses of parametric modeling in architectural design. We used Architecturology as scientific support of the research. Architecturology provides a scientific language for describing cognitive operations of architectural design. This language aims to interrogate conception through cognitive operations of giving measurements to an artifact.

In this reseach, we distinguished operations that give measurement to an architectural object and operations that give measurement to a parametric model's object.

We found out different kind of operations implied in parametric model conception:

-elementary operation of conception (*slicing, dimensioning* and *referencing*) and;

-pragmatic operation of conception (*interpretating, pooling* and *translating in parametric geometry*).

These operations show that we can distinguish conception of parametric modelling and architectural design, but it also show that these operations are intricate. Relevancies and references implied in elementary operations of conception are porous. This porosity between parametric modelling and architectural design is allowed by the pragmatic operations and by the construction of a common knowledge.

Thus the theoretical framework developed to analyze parametric architectural design situations leaded us to analyze various methodological or technical support tools proposed for assisting parametric modeling. For example, we analyzed the different supports provided by the large community of parametric modeling users and the "patterns" method developed by R. Woodbury.

Thanks to the different results of the research (cognitive operation identified and existing helps analyzed) we proposed didactics tools for assisting architects to parametric design.

**Key words: Architecturology, parametric modeling, architectural conception, pedagogy**



|   |     |
|---|-----|
| Introduction .....  | 9   |
| 1 Description du contexte de la recherche.....  | 12  |
| 1.1 Description du terrain de recherche à partir des discours existants .....   | 13  |
| 1.2 Etats de l'art sur les usages du numérique et de la modélisation paramétrique en architecture<br>16   |     |
| 1.3 Construire une connaissance de la conception.....   | 19  |
| 2 Les mécanismes informatiques définissant les modeleurs paramétriques.....   | 21  |
| 2.1 Les modeleurs paramétriques du point de vue de leurs mécanismes informatiques.....  | 23  |
| 2.2 Mises en œuvre de mécanismes paramétriques : caractéristiques de modeleurs .....  | 40  |
| 2.3 Analyse des modeleurs choisis .....   | 47  |
| 3 Etat des pratiques de la modélisation paramétrique en agence .....  | 63  |
| 3.1 Modélisation paramétrique et conception architecturale : quelles pratiques ?.....   | 65  |
| 3.2 Pratiques de consultation externes : le cas de Gehry Technologies .....   | 70  |
| 3.3 Pratiques de consultation interne : le cas de Foster and Partners.....  | 92  |
| 3.4 La profession d'architecte : la modélisation paramétrique comme nouveau métier ou comme<br>extension de l'expertise de l'architecte ? .....         | 100 |
| 4 Les aides à la modélisation paramétriques existantes .....  | 105 |
| 4.1 Une aide à la modélisation paramétrique en architecture : La méthode des Patterns .....   | 106 |
| 4.2 Supports en ligne : informations en libre accès .....   | 122 |
| 5 Construction de la méthode d'analyse et corpus rassemblé .....  | 141 |
| 5.1 Appareillage théorique et objet de recherche.....   | 143 |
| 5.2 Les entretiens .....  | 154 |
| 5.3 Les observations participantes .....  | 166 |
| 5.4 Les expérimentations .....  | 170 |
| 5.5 Synthèse du corpus .....  | 181 |
| 6 Analyse et caractérisation des opérations cognitives en jeu lors de l'usage de la modélisation<br>paramétrique pour la conception architecturale..... | 193 |
| 6.1 Conception de modèles paramétriques : Opérations cognitives élémentaires .....  | 195 |
| 6.2 Spécificité de la mise en œuvre des opérations conception de modèles paramétriques.....   | 204 |
| 6.3 Relations entre opérations de conception du modèle paramétrique et opérations de conception<br>architecturale.....                                  | 210 |
| 6.4 Opérations tierces impliquées .....   | 218 |
| 6.5 Synthèse des caractérisations cognitives de la conception de modèles paramétriques en<br>conception architecturale .....                            | 231 |
| 7 Analyse architecturologique de l'usage de la méthode des Patterns .....   | 235 |

|          |   |     |
|----------|---|-----|
| 7.1      | Expérimentations menées .....   | 236 |
| 7.2      | Spécificités cognitives de l'usage de Samples pour la conception de modèles paramétriques en conception architecturale.....         | 244 |
| 7.3      | Discussion de l'usage des patterns et des samples comme aide à la modélisation paramétrique pour la conception architecturale ..... | 257 |
| 8        | Proposition d'une pédagogie de la modélisation paramétrique en conception architecturale  | 260 |
| 8.1      | Objectifs pédagogiques de l'apprentissage de la modélisation paramétrique par des architectes .....                                 | 261 |
| 8.2      | Supports didactiques pour l'apprentissage de la modélisation paramétrique en conception architecturale.....                         | 274 |
| 8.3      | Support pédagogique, pour un enseignement de la modélisation paramétrique en conception architecturale.....                         | 287 |
|          | Conclusion.....   | 312 |
|          | Bibliographie .....   | 316 |
| Annexes: |   |     |
| 1        | Retranscriptions d'entretiens.....  | 327 |
| 2        | Analyses architecturologiques.....  | 340 |
| 3        | Catalogue de samples construit .....  | 417 |
| 4        | Retour sur les enseignements so823-2011 et so823-2012.....  | 475 |
| 5        | Supports en ligne d'aide à la modélisation paramétrique.....  | 481 |

## Introduction

Cette thèse a été menée au sein du laboratoire MAP-maacc de l'École Nationale Supérieure d'Architecture de Paris la Villette. Elle a bénéficié d'un financement du Ministère de la Culture et de la Communication français par le moyen d'une Allocation d'Etude Spécialisée de trois ans.

Cette recherche s'intéresse aux usages des techniques dites « de modélisation paramétrique » pour la conception architecturale. La modélisation paramétrique a été développée à l'origine pour la conception dans les secteurs de l'aéronautique et de l'automobile. Son appropriation relativement récente pour la conception en architecture pose question. Un modèle paramétrique est construit à partir d'un ensemble d'éléments (géométriques, numériques, *etc.*) mis en relation dans un système de dépendances (R. Aish & Woodbury 2005; Woodbury 2010). Ce système de dépendances peut être visualisé sous une forme symbolique (celle d'un graphe ou d'un tableau par exemple, en fonction des modeleurs) (Agbodan 2002). Un modèle paramétrique n'a pas de forme ou de contenu figé, il peut être modifié à tout moment à partir des variables indépendantes, appelées paramètres, qui le définissent (Maculet & Daniel 2005). Quand un de ces paramètres est modifié, le système de dépendance constituant le modèle paramétrique est alors mis à jour, générant une nouvelle géométrie. La modélisation paramétrique permet ainsi d'explorer un grand nombre de variantes d'un objet à concevoir. En cela, ce mode de représentation est potentiellement un outil très puissant pour la conception architecturale (Woodbury 2010). Ces variantes de l'objet à concevoir sont délimitées par un espace de transformations déterminé par le concepteur lorsqu'il construit le modèle.

Cette recherche interroge l'usage de la modélisation paramétrique pour la conception architecturale en termes d'opérations cognitives de conception. L'objet de recherche de cette thèse est la caractérisation des opérations cognitives impliquées lors de processus de conception architecturale impliquant la modélisation paramétrique. Cette spécification vise à définir les connaissances et les mécanismes impliqués par ces processus afin d'en assister l'apprentissage par des architectes.

L'objectif de la recherche est double, il est #1- de construire une connaissance des mécanismes cognitifs en jeu lors d'usages de la modélisation paramétrique en conception architecturale, et #2- de construire une aide à la mise en œuvre de ces mécanismes par des outils pédagogiques et didactiques.

L'appareillage théorique construit par l'architecturologie pour connaître la conception en termes d'opération d'attribution de mesures (Boudon 2004; Boudon et al. 2000) est le support scientifique de ce travail. C'est à partir de l'architecturologie que la problématique de cette recherche a été construite:

**« Quelles sont les opérations cognitives de conception impliquées lorsque la conception de modèles paramétriques est utilisée en conception architecturale ? Quelle pédagogie proposer pour accompagner l'apprentissage de ces opérations? ».**

Cette thèse propose dans un premier temps d'expliciter le contexte des interrogations soulevées par la thèse ainsi que différents états de savoir. Ce texte situe la problématique de la thèse par rapport aux différents discours existants sur les usages de la modélisation paramétrique en architecture. Cela nous permet de situer notre approche, sur la conception du point de vue de ces opérations cognitives (Chapitre 1). Un état du savoir sur les modeleurs paramétriques impliqués en architecture aujourd'hui est également effectué. Cet état du savoir permet de situer les modeleurs du point de vue informatique et de pointer de premières difficultés et particularités de leurs usages (Chapitre 2). Suite à cet état des modeleurs paramétrique, le Chapitre 3 rend compte de l'état des pratiques actuelles de la modélisation paramétrique en agence d'architecture. Enfin, le Chapitre 4 présente un état des aides à la pratique de la modélisation paramétrique. La méthode des « Patterns » proposée par Robert Woodbury (R. Aish & Woodbury 2005; Marques & Woodbury 2006; Qian et al. 2007; Woodbury 2010) pour l'aide à la modélisation paramétrique est interrogée ainsi que, les aides à la modélisation paramétrique, fournies sur le web, sont également interrogées. Les richesses, les potentialités, mais aussi les limites de ces aides sont abordées.

Dans un second temps, la thèse expose l'appareillage théorique et méthodologique mis en œuvre dans la recherche. Le Chapitre 5 présente les méthodes d'analyse et de recueil de données. La méthode d'analyse s'ancre au sein de l'Architecturologie Appliquée (Lecourtois 2012) et, les données sont recueillies grâce à des méthodes d'entretien, d'observation participante et d'expérimentation. Dans ce chapitre, le corpus de données effectivement construit est également décrit. Ce corpus se constitue à la fois de processus de conception mis en œuvre par des architectes dans des contextes professionnels (entre autres au sein des agences Gehry Partners, Foster and Partners et Ateliers Jean Nouvel) et, de processus de conception mis en œuvre par des étudiants architectes, mis en œuvre dans des contextes d'apprentissage de la modélisation paramétrique (issues de plusieurs Ecoles d'Architecture françaises et étrangères).

Dans un troisième temps la thèse présente les résultats issus des analyses menées. Le Chapitre 6 rend interroge les spécificités des opérations cognitives impliquées dans la modélisation paramétrique, ainsi que les relations entre conception de modèles paramétriques et conception architecturale. Les compétences et mécanismes cognitifs sollicités, en particulier les mécanismes logiques et de collaboration, sont explicités. Dans le Chapitre 7, les spécificités cognitives particulières à l'usage d'aides à la modélisation paramétrique pour la conception de modèles paramétriques en conception architecturale sont exposées. Ces analyses montrent l'usage de la modélisation paramétrique nécessite la mise en œuvre d'opérations cognitives spécifiques et parfois inhabituelles pour un architecte. Ces opérations cognitives demandent des compétences spécifiques en géométrie et en programmation. Une pédagogie spécifique à l'apprentissage de la modélisation paramétrique en conception architecturale semble donc nécessaire.

Finalement, le Chapitre 8 interroge et propose des ressources pour l'apprentissage de la modélisation paramétrique pour l'architecture. Savoirs, savoir-faire et savoir-être nécessaires sont spécifiés. Des objectifs pédagogiques pertinents au contexte actuel de ces pratiques sont définis (Chapitre 8). Des supports didactiques et pédagogiques sont proposés et construits, ils constituent un programme pédagogique spécifique à l'enseignement de la modélisation paramétrique en école d'architecture et, des supports didactiques donnant accès à un « espace des savoirs de la modélisation paramétrique ». Ces supports se composent d'une plateforme didactique [parametric-ressources.com] de



publication de ressources techniques associées à la méthode des Patterns de Robert Woodbury ainsi que, d'une plateforme didactique de publication de ressources théoriques [dnarchi.fr].

# 1 Description du contexte de la recherche

Ce chapitre propose de positionner la recherche par rapport aux discours existants sur les usages de la modélisation paramétriques en conception architecturale (1.1 et 1.2) et situe l'approche scientifique choisie pour construire une connaissance de la conception : l'architecturologie (1.3).

## Plan du chapitre :

|            |  |           |
|------------|--|-----------|
| <b>1.1</b> | <b>Description du terrain de recherche à partir des discours existants .....</b>                           | <b>13</b> |
| 1.1.1      | Modélisation paramétrique et architecture : usages pour la recherche patrimoniale et l'histoire            | 13        |
| 1.1.2      | Modélisation paramétrique et prospective au sein de l'architecture   | 13        |
| <b>1.2</b> | <b>Etats de l'art sur les usages du numérique et de la modélisation paramétrique en architecture .....</b> | <b>16</b> |
| <b>1.3</b> | <b>Construire une connaissance de la conception .....</b>  | <b>19</b> |

## **1.1 Description du terrain de recherche à partir des discours existants**

Les pratiques interrogeant conjointement la modélisation paramétrique et l'architecture sont diverses. Nous proposons dans cette première partie du Chapitre 1 d'en situer les principales, ceci en vue de positionner et de circonscrire celles qui nous intéressent dans cette recherche : les pratiques de la modélisation paramétrique en conception architecturale.

Nous esquissons ici les contours de deux pratiques majeures de la modélisation paramétrique en architecture : les explorations à visées scientifiques et patrimoniales (1.1.1) et, les explorations menées au sein de la pratique architecturale, en particulier avec des visées techniques et prospectives (1.1.2).

### **1.1.1 Modélisation paramétrique et architecture : usages pour la recherche patrimoniale et l'histoire**

Un des premiers usages de la modélisation paramétrique pour l'architecture a été effectué par Mark Burry et son équipe du SIAL (Spatial Information Architecture Laboratory) dès les années 90. La modélisation paramétrique est alors utilisée dans le cadre d'une recherche visant à restituer la Sagrada Familia d'Antonio Gaudi malgré les lacunes de la documentation disponible (J. Burry & M. Burry 2006). En effet, le chantier de la Sagrada Familia n'a pas été terminé du vivant de son architecte et la plupart des documents décrivant l'ouvrage (plans, maquettes, etc.) ont été détruit au début du XXe. Les recherches du SIAL visent à proposer des reconstructions cohérentes avec les règles de génération de géométries identifiables dans le projet à partir des photographies et des images fragmentées disponibles (J. Burry & M. Burry 2006). La modélisation paramétrique permet alors de modéliser des familles de formes (colonnes, etc.) architecturales, visant à être évaluées, ici du point de vue de la cohérence du projet.

De la même façon, la modélisation paramétrique est utilisée pour interroger les propriétés géométriques de différents ouvrages, existants ou détruits. Les modèles paramétriques développés permettent alors de confirmer ou d'infirmer les hypothèses de travail, par exemple les proportions des églises byzantines (Potamianos et al. 1995) ou les propriétés structurelles de certains ouvrages gothiques (Nikolinakou et al. 2005).

Ici la modélisation paramétrique est utilisée dans le champ de l'architecture pour des visées patrimoniales et scientifiques de reconstitution. Ces recherches relèvent d'une approche technique et scientifique, mais ne permettent pas d'aborder la question de l'usage de la modélisation paramétrique en conception architecturale.

### **1.1.2 Modélisation paramétrique et prospective au sein de l'architecture**

Des architectes se sont appropriés la modélisation paramétrique depuis de nombreuses années et en explorent les potentialités techniques et architecturales. Ces usages ont parfois donné lieu à une littérature constituée de descriptions et de retours réflexifs sur des pratiques de projets (Shelden 2002;

Cache 1997; Landa & Spuybroek 2005) voire, de discours doctrinaux aux visées larges, comme ceux sur le « Parametricism » de Patrick Schumacher, proposant une relecture de l'histoire de l'architecture sous l'angle de la modélisation paramétrique (Schumacher 2008). La part de cette littérature plus spécifiquement orientée vers la description de pratiques (Shelden 2002; Cache 1997; Smith 2005) constitue un support d'analyse riche pour aborder la question de l'usage de la modélisation paramétrique en conception architecturale (cf. Chapitre 5). On peut observer certains points récurrents dans les pratiques de la modélisation paramétrique en conception architecturale. Ces points sont esquissés dans les paragraphes suivant.

#### **- Modélisation paramétrique et prouesses techniques, géométriques et structurelles**

Dans la littérature rendant compte des usages de la modélisation paramétrique pour l'architecture, la question de l'innovation technique est souvent centrale. C'est le cas dans la pratique de rationalisation de structure ou de calepinage de façade comme peut le mettre en œuvre Cecil Balmond (Balmond 2007) ou encore, les membres des bureaux d'étude RFR (Schiftner et al. 2012) et Buro Happold (Imbert et al. 2012).

#### **- Modélisation paramétrique et partage de données, la question du « BIM »**

De nombreux discours sont produits sur les BIM (Building Information Modelling) (Eastman et al. 2008). Ces discours mêlent parfois la modélisation paramétrique à la question plus large de la maquette numérique comme support pour le partage d'informations. Le BIM est une méthode de construction de maquette numérique s'intéressant à la possibilité de rassembler, dans un même modèle, des informations, multiples propres aux différents corps de métier intervenant dans le projet. Les informations partagées sur la maquette (propriétés, géométries, droits d'accès, documents textuels, etc.) peuvent alors permettre, à différents acteurs, de travailler ensemble sur un même support (Kiviniemi in Kocaturk & Medjdoub 2011, p.125). Certains modeleurs paramétriques peuvent se prêter à une utilisation « BIM » (Eastman et al. 2008), nous l'aborderons plus précisément dans la section 2.2 de cette recherche.

#### **- Modélisation paramétrique et fabrication à commande numérique**

Dans de nombreux discours, la modélisation paramétrique est associée à l'usage de machines de production à commandes numériques. Par exemple l'agence Objectile revendique la mise en place d'un « continuum » de la conception à la fabrication grâce à l'usage de modeleurs paramétriques et de machines à commande numérique (Cache 1997). Les potentialités des machines à commande numérique (Robots, découpe laser, etc.) sont multiples (Marin 2010, p.73-78; Malé-Aleman 2009). Les enjeux de la production de pièces toutes différentes grâce à des machines à commande numérique ont été pointés, entre autres, par Frédéric Migayrou lors de son commissariat pour l'exposition *Architecture non-standard* (Migayrou 2003). Si cet enjeu est fort (à en voir de nombreux thèmes de colloques récents comme par exemple "Matérialités Contemporaines" (Liveneau & Marin 2012), ou TexFab [tex-fab.net]), les technologies mises en œuvre restent en cours de développement et encore peu généralisables (Picon 2012). A ce sujet, le champ de la recherche reste ouvert et vaste. Nous n'interrogerons pas ces enjeux dans la présente recherche.

#### **- Modélisation paramétrique et biomorphisme**

Dans de nombreux projets d'architecture comme dans le développement d'outils (Rutten 2010) ou encore dans la théorisation de l'architecture contemporaine (Neil 2002), on observe le recours à de

nombreux processus que Philippe Marin, dans sa recherche « Implication des mécanismes évolutifs appliqués à la conception architecturale », nomme « bio inspirés » (Marin 2010, p.90). Ces explorations techniques et prospectives sont riches et parfois liées à l'usage de la modélisation paramétrique. Elles ne seront cependant pas développées dans notre recherche.

**- Terrain de la présente recherche :**

La recherche développée dans cette thèse s'intéresse aux usages de la modélisation paramétrique en architecture, spécifiquement en situation de conception architecturale. Ces situations de conception architecturale peuvent être entremêlées avec des enjeux de fabrication ou de collaboration, c'est ce que nous verrons dans le Chapitre 3. Néanmoins, notre terrain de recherche se constitue spécifiquement de **cas d'usages de modèles dits paramétriques en conception architecturale.**

## **1.2 Etats de l'art sur les usages du numérique et de la modélisation paramétrique en architecture**

Les connaissances formulées sur notre terrain portent, pour une part, sur l'usage du numérique en général en architecture et, d'autre part, sur l'usage de la modélisation paramétrique en architecture.

### **- Le point de vue historique**

Une littérature donne un éclairage historique à l'usage de technologies numériques en conception architecturale. Cet éclairage permet de percevoir les ramifications des technologies numériques et leurs enjeux dans l'architecture (Picon 2010) ainsi que le succès critique de l'architecture numérique (Carpo 2012). Ces travaux permettent de situer plus largement les pratiques de la modélisation paramétrique dans le contexte du rapport de l'architecture aux technologies numériques. Ils permettent également de situer l'évolution de ces rapports. Néanmoins, ces approches historiques ne permettent pas de connaître les spécificités cognitives impliquées lors de l'usage de technologies numériques pour la conception architecturale.

### **- Les points de vue philosophiques / sur le rapport entre l'homme et la technique en général**

Les points de vue philosophiques sur la technique sont multiples et nous aident aujourd'hui à penser les rapports de la technique aux activités humaines. En particulier les apports, distincts, de Gilbert Simondon et Bernard Stiegler nous paraissent majeurs. Gilbert Simondon permet de penser le rapport à la technique en dépassant la dichotomie homme / technique, en pensant avant tout la relation qu'ils entretiennent et qui constitue un processus d'*individuation* donnant sa réalité à l'*individu technique* (Simondon 1958; Barthelemy 2008). Les apports de Gilbert Simondon sont interrogés dans le cadre de recherches en cours pour penser l'architecture numérique et le rapport de l'architecte à l'outil numérique (Bourbonnais 2010). Stiegler, quant à lui, permet d'interroger la création mais aussi l'enseignement à l'aune de la société contemporaine et de ces enjeux techniques liés au numérique (Stiegler et al. 2012). Néanmoins, ces travaux ne nous permettent pas de questionner l'usage de la modélisation paramétrique en conception architecturale du point de vue des activités cognitives mises en oeuvre.

### **- Les points de vue philosophiques / sur le rapport entre l'architecture et les technologies numériques en particulier**

Plus précisément sur l'architecture numérique, Kostas Terzidis dans *Algorithmic Architecture* interroge à la fois les potentialités techniques de l'algorithmique et son rapport à l'architecture d'un point de vue philosophique (Picon in Terzidis 2006). Dans cet ouvrage, Terzidis montre l'ordinateur non comme une extension des facultés de l'architecte, mais comme un partenaire radicalement « autre », faisant osciller la pratique de l'architecte entre la spontanéité de ce qu'il nomme la création et la rigueur de son explicitation dans un algorithme (Terzidis 2006). Dans le cadre de cette thèse, nous pointons également cette oscillation (cf. Chapitre 6) mais non sous un angle philosophique.

Plus spécifique à l'architecture numérique, on peut également relever les apports du philosophe Stanford Kwinter et du critique Jeffrey Kipnis pour une compréhension à la fois globalisante et

spécifique à l'architecture contemporaine à l'aune des technologies numériques (Kipnis in Di Cristina 2001; Kwinter 2004).

Par ailleurs, l'apport de Gilles Deleuze sur certaines notions est important. Ces apports nous permettent de mettre en perspective les notions d'instance et d'espace de solutions (cf. Chapitre 2) *via* les concepts d'*Objectile* (Deleuze 1988) et de *différence* (Deleuze 1968).

Si ces différents apports sont majeurs et nous aident à mettre en perspective notre analyse des processus cognitifs de conception, impliqués lors de l'usage de technologies numériques pour la conception architecturale, ils ne permettent pas d'en construire une connaissance scientifique.

### **- Les points de vue sociologiques : une transformation des pratiques et des organisations des agences**

Des recherches issues des champs de la sociologie portent sur la transformation des pratiques de l'architecture en lien avec les technologies numériques. C'est le cas des recherches de Guy Tapie dont l'ouvrage *Les architectes : Mutations d'une profession* (2000), rend compte. C'est également le cas du travail de Stéphane Hanrot dans son texte « Les enjeux de la maîtrise d'œuvre, projet et technologie », publié sous la direction de Jean-Jacques Terrin dans l'ouvrage *Maîtres d'ouvrage, maîtres d'œuvre et entreprises, De nouveaux enjeux pour les pratiques de projet* (2005).

Dans ces ouvrages, certaines caractéristiques des pratiques des architectes induites par l'usage de technologies numériques sont pointées. Ces caractéristiques sont :

- la transformation des pratiques, avec en particulier l'augmentation de l'archivage et de la traçabilité des actions de chacun et son impact sur le contrôle et la responsabilisation de l'architecte (Tapie 2000, p.89) ;

- la transformation de l'organisation des agences, avec l'introduction d'outillages pour le partage de données (armoires à plans, serveurs) impliquant de nouvelles méthodes de dessin pour le partage de données compatibles (Tapie 2000, p.87);

- ainsi que l'apparition de nouveaux postes spécialisés dans l'informatisation de l'agence (support informatique), la manipulation d'outils informatiques spécifiques (pour le dessin, le rendu d'images, etc.) (Tapie 2000, p.94) ou encore pour la coordination de la synthèse du modèle (Tapie 2000, p.116);

- mais aussi la transformation du cadre juridique de la pratique de l'architecture, visant à encadrer l'introduction de documents informatiques (comme la maquette numérique) et le partage de données dans le cadre contractuel de la pratique de l'architecte (Hanrot in Terrin 2005; Tapie 2000, p.105).

Ces éléments de connaissances ne portent pas spécifiquement sur la conception architecturale mais plus globalement sur les pratiques menées en agence et ils ne portent pas non plus précisément sur les usages de la modélisation paramétrique. Néanmoins ces apports sont importants pour appréhender les pratiques de la modélisation paramétrique car ils pointent des transformations globales des pratiques des architectes.

### **- Recherches sur les usages de la modélisation paramétrique**

Des chercheurs interrogent spécifiquement l'usage de la modélisation paramétrique en conception architecturale. Néanmoins, il apparaît que ces recherches pointent les spécificités techniques de celle-ci

ou bien, interrogent des objets autres que la conception tels que la collaboration (Davis et al. 2011) ou la création (Chien & Yeh 2012).

Les apports plus spécifiques de Robert Woodbury et son équipe sur les usages de la modélisation paramétrique en architecture sont exposés plus précisément dans le Chapitre 4.1.

Finalement, si tous ces points de vue construisent des connaissances fortes et complémentaires sur les usages de technologies numériques en conception architecturale, on se rend compte ici qu'ils ne permettent pas d'aborder spécifiquement les activités cognitives impliquées lors de l'usage de la modélisation paramétrique pour la conception architecturale (cf. Tableau 1).

| <i>littérature</i>   | <i>Apports pour la thèse</i>   | <i>Limites pour la thèse</i>   |
|--|--|--|
| Retours descriptifs et réflexifs par des praticiens (architectes, ingénieurs, enseignants, développeurs, etc.) sur des cas d'usage de la modélisation paramétrique pour la conception architecturale | Descriptions de pratiques utilisées comme matériaux de corpus              | → L'objet de ces approches n'est pas les activités cognitives impliquées lors de l'usage de la modélisation paramétrique pour la conception architecturale |
| Histoire de l'architecture et des technologies numériques  | Mise en perspective des résultats de la thèse                              |  |
| Philosophie des rapports entre homme et technique, ou plus spécifique aux rapports entre architecture et techniques numériques   |  |  |
| Sociologie   | Apports sur les pratiques de l'architecture et les technologies numériques |  |

**Tableau 1 : apports et limites des approches historiques, philosophies et sociologiques pour la thèse**



### 1.3 Construire une connaissance de la conception

Nous venons de le voir, de nombreux travaux ont été effectués sur les usages des outils numériques et de la modélisation paramétrique (cf. 1.2) mais ces travaux n'interrogent pas spécifiquement ces usages pour la conception d'un point de vue cognitif.

Pour cette recherche, nous nous appuyons sur l'« architecturologique » pour construire une connaissance de l'activité cognitive de conception impliquée lors de l'usage de la modélisation paramétrique en conception architecturale.

**Distinguer <concevoir l'architecture> et <connaître scientifiquement la conception architecturale>.**

L'objet de recherche de l'approche architecturologique, est l'activité cognitive de conception architecturale.

En 1968 en France, l'enseignement de la conception architecturale par des ateliers verticaux dirigés par des maîtres est fortement critiqué et, une refonte de l'enseignement de l'architecture est mise en place (Violeau 2005). Entre autres questions soulevées, celle d'une possibilité de construire des connaissances scientifiques de la conception architecturale est abordée. Cette question épistémologique est l'enjeu de l'ouvrage *Sur l'espace architectural* de Philippe Boudon (1971). Dans cet ouvrage, Boudon expose la nécessité d'une distinction entre la construction d'une connaissance scientifique de la conception architecturale d'un côté et, la pratique de l'architecture de l'autre. Cette distinction relève d'une *rupture épistémologique*, telle que définie par Gaston Bachelard et peut être rapprochée d'un *obstacle épistémologique* (Bachelard 1934)<sup>1</sup>. Le positionnement de la recherche scientifique proposée relève d'une approche analytique (Engel & Dutant 2005; Dutant 2010). La construction de telles connaissances scientifiques se distingue des pratiques dites de « recherches par le projet ». La « recherche par le projet » propose, de la part de praticiens (architectes ou artistes), un retour réflexif sur leurs propres pratiques, sans postuler la nécessité d'une *rupture épistémologique* entre la pratique de ces praticiens et la connaissance empirique qu'ils en ont d'un côté et la construction d'une connaissance scientifique de l'autre.

Au moyen de ce positionnement scientifique qu'il postule, Philippe Boudon propose d'interroger la conception architecturale sous la forme d'*opérations cognitives d'attribution de mesures* (Boudon 1971). Il propose d'aborder ces opérations grâce au concept d'« échelles », dans un champ scientifique qu'il nomme « *architecturologie* ». On peut noter ici que ce n'est pas toute la pratique du projet architectural qui est interrogée (cette dernière associant de multiples activités, comme la représentation, la négociation, l'organisation des tâches, *etc.*), mais un aspect de celle-ci : l'élaboration de l'objet architectural, abordée comme une activité d'*attribution de mesures* (Boudon et al. 2000, p.78)<sup>2</sup>.

---

<sup>1</sup> Pour construire des connaissances scientifiques, il faut pouvoir couper avec les idées préconçues, l'expérience première, la connaissance générale, les termes mal définis, *etc.* : éléments que Bachelard a soulevé en termes d'*obstacles épistémologiques* (Bachelard 1934).

<sup>2</sup> « Si le projet désigne à la fois l'activité de production graphique, la mise en commun des compétences entre les différents acteurs de l'agence et le travail de négociation avec les commanditaires et exécutants, l'*architecturologie* se propose de cerner plus précisément le travail de mise en forme proprement dit, c'est-à-dire

Les questionnements épistémologiques soulevés et l'appareillage théorique proposé par Philippe Boudon à partir des années 70 ont été développés par lui-même ainsi que par d'autres chercheurs, en particulier au sein du laboratoire URA-CNRS le LAREA<sup>3</sup> (Laboratoire d'Architecturologie et de Recherches Epistémologiques sur l'Architecture).

L'approche scientifique nommée *architecturologie* vise donc à construire une connaissance de la conception architecturale et non à concevoir l'architecture ou à améliorer une production architecturale (cf. Tableau 2).

|                |                            |  |
|----------------|----------------------------|--|
| <i>Champ :</i> | Architecture               | Architecturologie  |
| <i>Objet :</i> | Concevoir une architecture | Construire une connaissance de l'activité cognitive de conception architecturale |
| <i>Visée :</i> | Construire, concevoir,...  | Connaître scientifiquement   |

**Tableau 2 : Distinctions entre architecture et architecturologie proposées par Philippe Boudon dans *Sur l'espace architecturale* (1973)**

### Intérêt de l'approche quant aux questions de la recherche

L'architecturologie fondamentale propose un appareillage de concepts construits de manière *a priori* pour décrire la conception. Si les questionnements de l'architecturologie ont été initiés par une réflexion sur la conception architecturale, les concepts développés permettent d'interroger la conception en général, toutefois entendue comme opérations cognitives d'attribution de mesures (Boudon 2004). Dans le cadre de notre recherche, l'outillage scientifique et théorique de l'architecturologie nous permet d'interroger la conception architecturale et la conception de modèles paramétriques.

L'architecturologie dite appliquée (Lecourtois 2008; Lecourtois 2005) propose des méthodes procédant des concepts de l'architecturologie pour interroger et décrire des corpus de cas de conception. Ces méthodes permettent d'interroger la conception au travers de discours ou de productions graphiques (Lecourtois 2004; Lecourtois 2011; Ben Rajeb 2012). Dans le cadre de notre recherche, l'architecturologie appliquée nous permet d'interroger des situations concrètes d'usage de la modélisation paramétrique en conception architecturale.

L'architecturologie est parfois critiquée pour relever d'une approche purement analytique ou pour la relative difficulté du langage qu'elle propose (Girard 1995). Néanmoins, elle fournit des outils théoriques permettant de décrire toute opération d'attribution de mesures au travers de concepts construits et argumentés. Par ailleurs, les méthodes de l'architecturologie appliquée permettent de mettre ces concepts à l'épreuve de corpus de cas de conception et de questionner à la fois ces cas comme les concepts utilisés (Lecourtois 2011). Ces deux aspects de l'architecturologie font de cette approche une méthode scientifique paraissant pertinente pour répondre aux questions de cette recherche.

---

*le travail d'élaboration de l'objet architectural, un travail qui donne forme et mesure à l'espace.* » (Boudon et al. 2000, p.78).

<sup>3</sup> En 2009, le LAREA et le laboratoire ARIAM (Atelier de Recherche en Informatique et Architecture) se sont associés, pour former le MAACC (laboratoire de Modélisations pour l'Assistance à l'Activité Cognitive de la Conception) en 2012. Cette recherche s'inscrit dans ce laboratoire.

## 2 Les mécanismes informatiques définissant les modeleurs paramétriques

Après avoir situé le contexte de la recherche du point de vue du terrain étudié et de l'approche scientifique choisie, nous proposons dans ce chapitre d'interroger la modélisation paramétrique en vue d'en faire un état du point de vue des mécanismes informatiques qu'elle implique. Existente-t-ils des caractéristiques communes aux logiciels communément rassemblés sous le vocable de «modeleurs paramétriques», et si oui, quelles sont-elles ? Ces caractéristiques informatiques sont importantes à connaître car elles permettent d'établir des hypothèses sur l'usage de ces modeleurs. En effet, pour certains praticiens de la modélisation paramétrique (Shelden 2002, p.25)<sup>4</sup> ou certains philosophes de la technique (Simondon 1958; Barthelemy 2008) l'impact des spécificités de l'outil sur la conception semble inévitable.

Pour définir les caractéristiques des modeleurs paramétriques, nous étudions dans ce chapitre leurs mécanismes informatiques dont nous donnons une définition en 2.1. La section 2.2 de ce chapitre est consacrée à l'étude des principaux modeleurs utilisés en architecture (archiCad, Revit, Autocad, *etc.*) et à la comparaison des mécanismes informatiques qui les caractérisent. La section 2.3 propose une analyse détaillée des modeleurs Grasshopper, Generative Components, Digital Project et DesignScript.

---

<sup>4</sup> « *The operations enabled by a chosen tool guide the operator to make specific types of objects or products that the tool affords* » « *the “trace” of the tool is inevitably felt in the resulting designs.* » (Shelden 2002, p.25).

## **Plan du chapitre :**

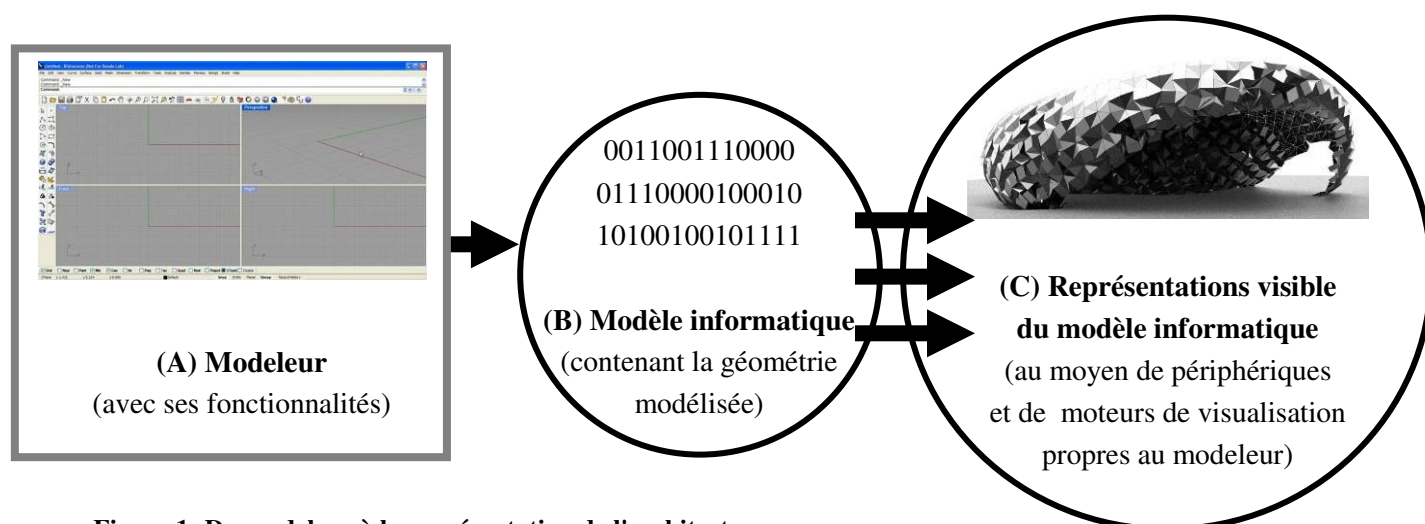
|            |  |
|------------|--|
| <b>2.1</b> | <b>Les modeleurs paramétriques du point de vue de leurs mécanismes informatiques... 23</b> |
| 2.1.1      | Modélisation paramétrique, une double représentation: explicite et symbolique<br>23        |
| 2.1.2      | Les approches paramétriques fonctionnelles et paramétriques équationnelles 28              |
| 2.1.3      | Les systèmes basés sur la propagation (« <i>Propagation Based Systems</i> ») 33            |
| <b>2.2</b> | <b>Mises en œuvre de mécanismes paramétriques : caractéristiques de modeleurs..... 40</b>  |
| 2.2.1      | Origines et évolution des modeleurs paramétriques 40                                       |
| 2.2.2      | Description de quelques uns des modeleurs paramétriques utilisés pour<br>l'architecture 40 |
| 2.2.3      | Méthode d'analyse de modeleurs choisis 45  |
| <b>2.3</b> | <b>Analyse des modeleurs choisis ..... 47</b>  |
| 2.3.1      | Grasshopper (GH) 47  |
| 2.3.2      | Generative Components (GC) 50  |
| 2.3.3      | Digital Project (DP) 52  |
| 2.3.4      | DesignScript (DS) 56   |

## 2.1 Les modeleurs paramétriques du point de vue de leurs mécanismes informatiques

### 2.1.1 Modélisation paramétrique, une double représentation: explicite et symbolique

**Modèle informatique d'une géométrie et représentations visibles de ce modèle.**

Un modeleur est un logiciel informatique (cf. Figure 1-A) permettant de construire un modèle géométrique numérique (cf. Figure 1-B), à partir duquel on produit différentes représentations: visualisées sur des écrans, imprimées, exportées dans des formats divers etc. (cf. Figure 1-C). Ces représentations peuvent renvoyer à des objets conçus ou en cours de conception (Quintrand 1990; Estevez 2001).



**Figure 1: Du modeleur à la représentation de l'architecture**

Pour la plupart des modeleurs utilisés pour l'architecture (Autocad, Rhinocéros, 3DS, etc.), lorsqu'un utilisateur crée un modèle numérique, il manipule une représentation géométrique de son modèle. Il visualise sa forme sur son écran et la génère ou la transforme par des opérations successives directement effectuées sur la forme. Dans la plupart des cas, le concepteur construit donc son modèle en manipulant directement la représentation géométrique de sa forme (cf. Figure 1-C) et non en manipulant le programme informatique générant cette forme (cf. Figure 1-B).

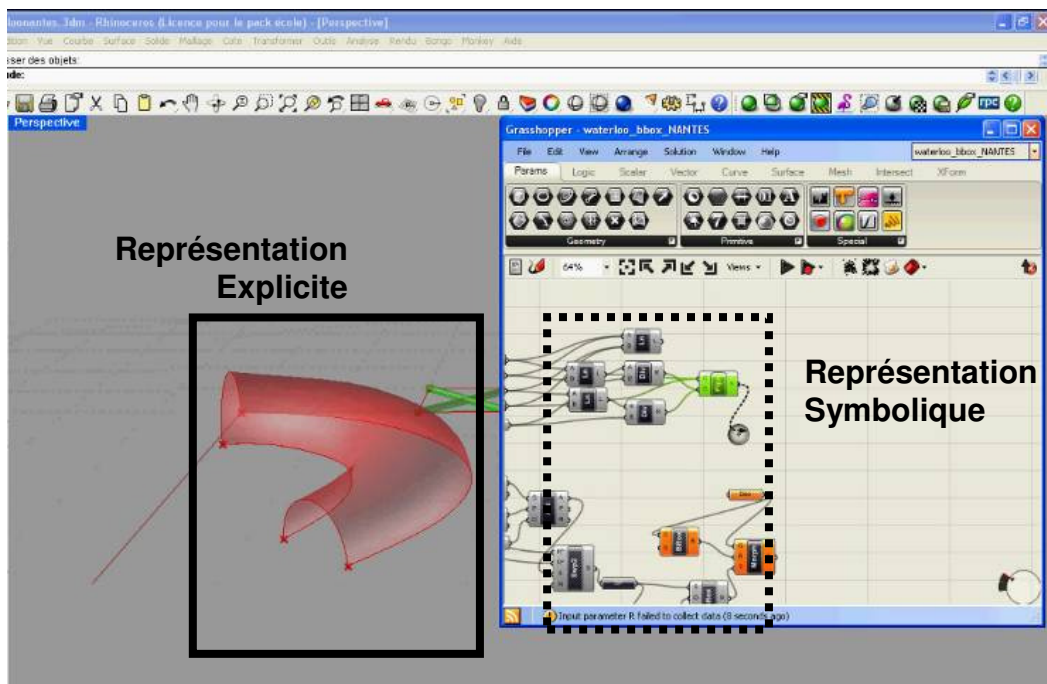
Or dans certains modeleurs, des éléments de ce programme informatique sont accessibles à l'utilisateur. L'accès, même partiel, à ce modèle informatique (cf. Figure 1-B) permet de générer facilement de multiples variantes d'une même géométrie. C'est ce qui caractérise les modeleurs paramétriques.

**Définition de la modélisation paramétrique : une modélisation permettant plusieurs représentations d'un modèle.**

Agbodan, dans sa thèse de doctorat en informatique (Agbodan 2002), propose une définition de la modélisation paramétrique relativement générale. D'autres proposent des définitions plus restrictives (Maculet & Daniel 2005; Lesage 2006; R. Aish & Woodbury 2005; Woodbury 2010; Hoffman & Joan-Arinyo 2005; Mineur 2009). Agbodan définit un modèle paramétrique comme « *une structure duale avec d'un côté les données et de l'autre le programme correspondant* » (Agbodan 2002, p.33). La dualité possède de multiples acceptions en mathématiques comme en géométrie, mais peut ici être entendue dans son sens commun de « *ce dit de propriétés qui vont par deux et qui présente un caractère de réciprocité* » (Rey-Debove & Rey 1992, p.582). Agbodan définit la modélisation paramétrique comme une structure mettant en relation deux ensembles d'éléments qui, sans être équivalents, renvoient à un même contenu : le modèle informatique. Ces deux éléments sont :

-d'un côté une représentation dite « **explicite** » d'un objet, c'est la représentation graphique de sa géométrie (cf. Figure 2 à gauche) et,

-de l'autre côté une représentation abstraite, dite « **symbolique** », du modèle informatique (cf. Figure 2 à droite). Cette dernière représentation est accompagnée d'algorithmes propres aux modélisateurs, permettant de recalculer le modèle *explicite* lorsque des éléments, de l'une ou l'autre représentation, sont modifiés (Agbodan 2002, p.33).



**Figure 2 : Exemple de représentations « explicites » et « symboliques » sur Grasshopper (plugin de Rhinocéros)**

La « représentation abstraite » ou « représentation symbolique » constitue une représentation informée du modèle, sur laquelle le concepteur peut intervenir pour transformer son objet. Une transformation locale sur l'une des deux représentations (*symbolique* ou *explicite*) du modèle paramétrique entraîne une mise à jour globale de celui-ci.

La représentation *explicite* (la visualisation géométrique de l'objet), quant à elle, est dite aussi « instance courante ». Pour Agbodan, elle est indispensable puisqu'elle permet de vérifier la cohérence du modèle (Agbodan 2002, p.37).

On peut s'interroger sur la relation de ces représentations : sont-elles équivalentes entre elles et renvoient-elles exactement aux mêmes informations (cf. Figure 3 -A)? Sont-elles équivalentes dans leur rapport au modèle informatique et permettent-elles une même interaction avec celui-ci (cf. Figure 3-B et -C) ?

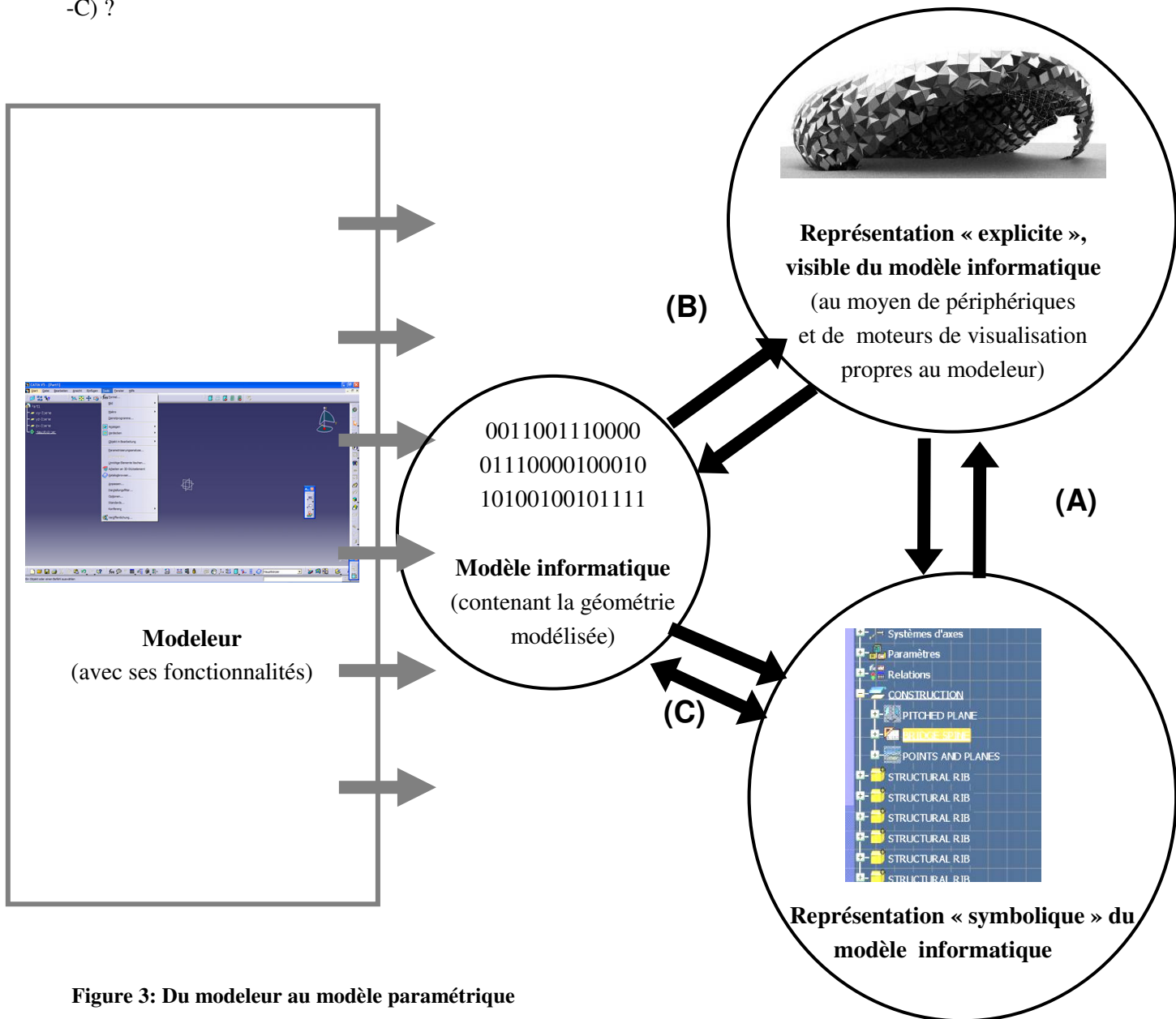
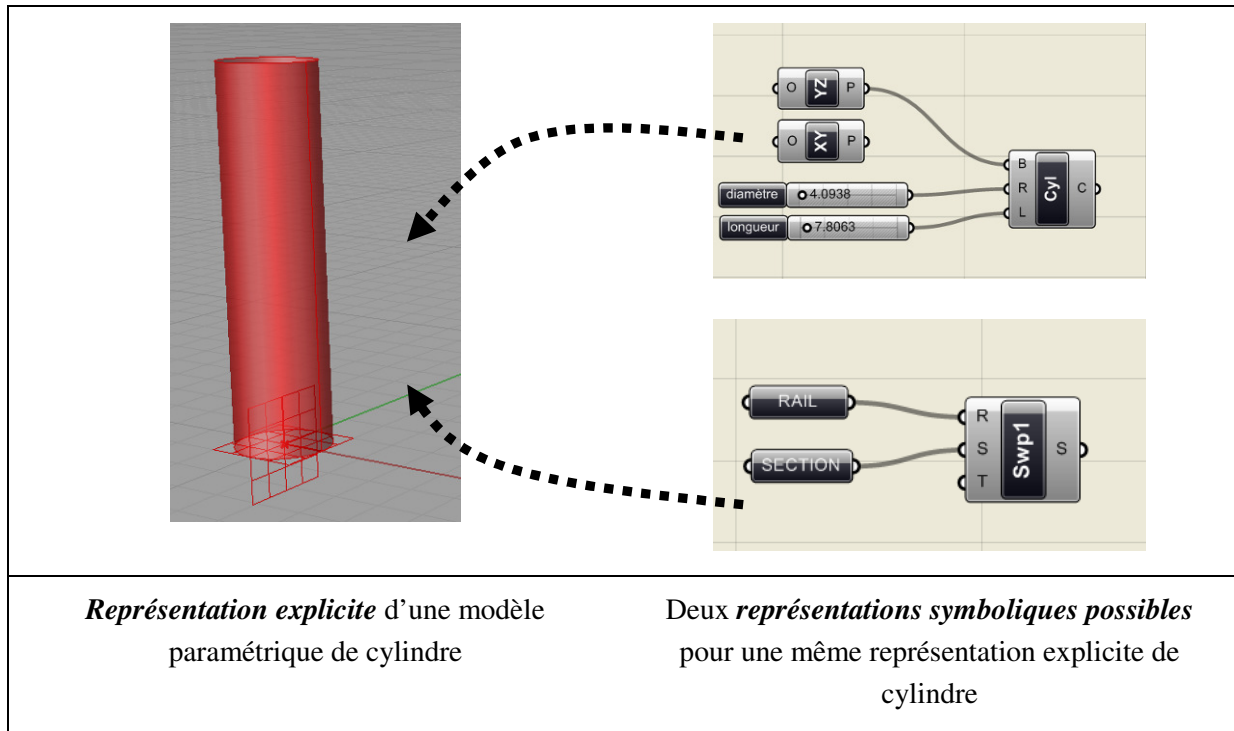


Figure 3: Du modeleur au modèle paramétrique

Représentation *explicite* et représentation *symbolique* peuvent donner accès à des informations différentes. Si la plupart du temps on peut reconstruire une figure *explicite* à partir d'une représentation *symbolique*, il est presque impossible de reconstruire une représentation *symbolique* à partir d'une représentation *explicite* : une multitude de représentations *symboliques* peuvent correspondre à une même représentation *explicite* (cf. Figure 4). Par exemple, pour une même géométrie de trame de cylindres, plusieurs représentations *symboliques* sont possibles : comme la translation d'un cylindre au

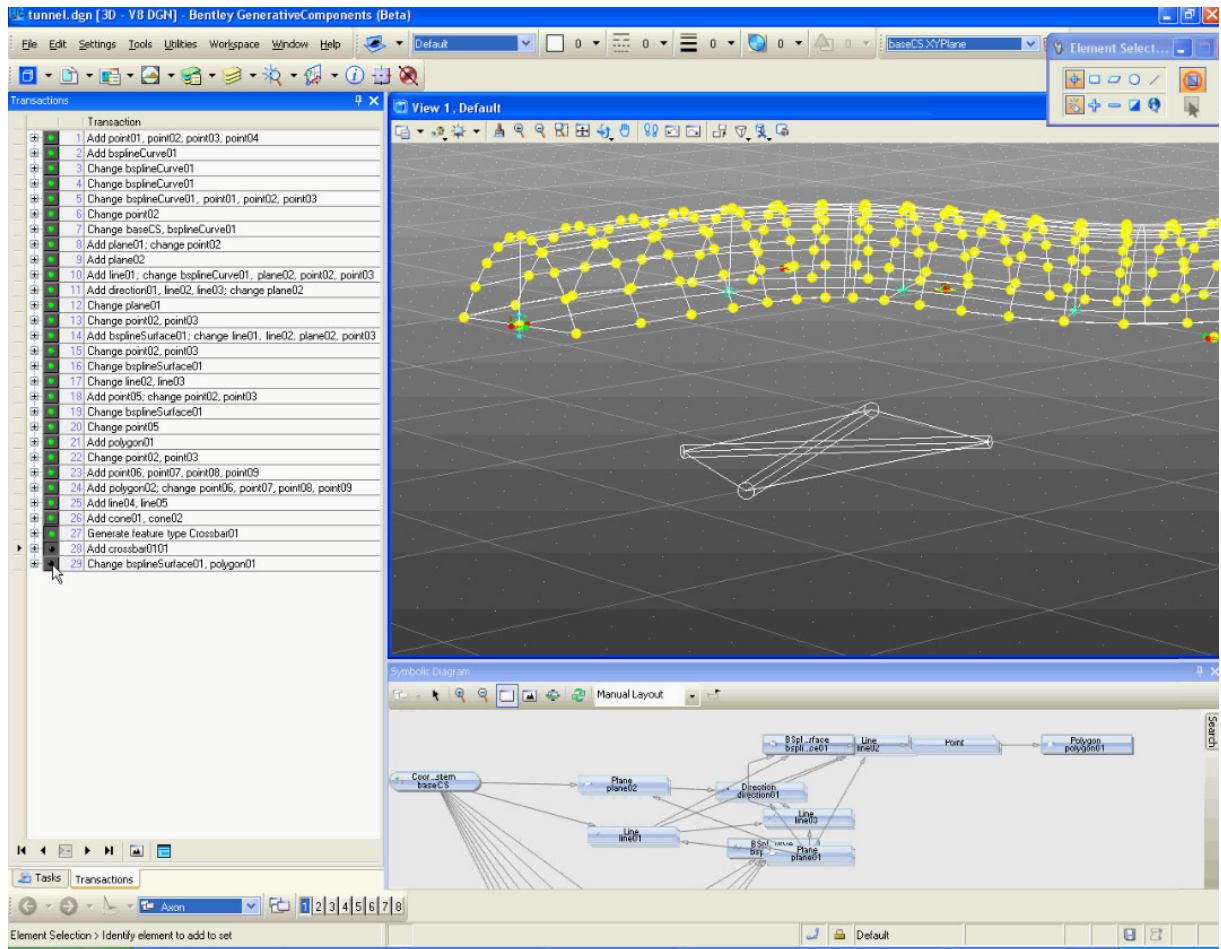
moyen d'une liste de vecteurs, la copie d'un cylindre au moyen d'une grille de point, *etc.* Le cylindre lui-même peut être représenté au moyen d'une primitive cylindre, de l'extrusion d'un cercle le long d'une courbe, ou encore de la révolution d'une droite autour d'un axe (cf. Figure 4). Finalement les relations entre représentations *symbolique* et *explicite* (cf. Figure 3-A) ne vont pas de soi dans le cas des modelers paramétriques utilisés en architecture.



**Figure 4 : Exemple de deux représentations *symboliques* possibles pour une même représentation *explicite* réalisées avec Grasshopper (utilisation d'une fonction générant une primitive cylindre en haut et d'une fonction « balayage sur un rail » en bas)**

Outre cette question de différence d'informations contenues dans les représentations *explicites* et *symboliques*, en fonction des modelers, les interactions possibles avec ces deux types de représentations (*explicites* et *symboliques*) ne sont pas les mêmes. Certains modelers autorisent des transformations uniquement sur le modèle *symbolique*, c'est le cas de Grasshopper, on ne peut modifier directement la représentation *explicite* du modèle (cela sera développé plus précisément en 2.3.1). D'autres modelers permettent plusieurs représentations *symboliques*, chaque représentation symbolique proposée au sein de ce même modeler donnant alors accès à des informations différentes, communiquées sous des visualisations différentes. C'est le cas du modeler Generative Components qui propose un graphe, ainsi qu'un tableau d'historique ou encore un accès au code décrivant le modèle (cf. Figure 5 et paragraphe 2.3.2).





**Figure 5 : Exemple de l'interface de Generative Components, avec une représentation explicite de la géométrie du modèle et, deux représentations symboliques du modèle, l'une par graphe (en bas) et l'autre par historique (à gauche)**

Au sein des modelleurs paramétriques, l'interaction de l'utilisateur avec le modèle informatique *via* la représentation *symbolique* et la représentation *explicite* est un sujet majeur (R. Aish & Woodbury 2005). La représentation *explicite* possède des traits de ressemblance directe avec l'objet représenté. La représentation *symbolique* est constituée de signes « symboliques » (c'est-à-dire renvoyant à des conventions extérieures (Everaert-Desmedt 1995, p.65)) parfois hybridés avec des signes « iconiques » (c'est-à-dire entretenant un lien direct que ce qu'ils représentent (Everaert-Desmedt 1995, p.53)). Les caractéristiques de ces représentations sont plus précisément abordées dans la section 2.3 consacrée à l'analyse des modelleurs. La représentation *symbolique* renvoie à des connaissances et des conventions spécifiques au modelleur. La lecture de ces signes est une des activités que nous visons à éclaircir dans cette thèse. Le recours des « tiers » nécessaire à la manipulation de signes symboliques propres aux modelleurs paramétriques sera une part importante du travail de construction d'une pédagogie de la modélisation paramétrique.

La suite de ce chapitre s'intéresse plus particulièrement aux mécanismes des représentations symboliques. Trois approches impliquées dans les mécanismes informatiques dont ils peuvent relever sont abordées ici : les approches paramétriques fonctionnelles et équationnelles (2.1.2), ainsi que l'approche par système basé sur la propagation (2.1.3).

## 2.1.2 Les approches paramétriques fonctionnelles et paramétriques équationnelles

Le fonctionnement des représentations *symboliques* peut se résumer ainsi : un concepteur conçoit une représentation *symbolique*, qui est ensuite traduite par le modelleur en représentation *explicite* (cf. Figure 6). Une représentation intermédiaire est alors construite par le modelleur. Cette représentation intermédiaire est plus ou moins accessible à l'utilisateur en fonction des modelleurs, cela sera abordé plus précisément en fonction des modelleurs étudiés dans la suite du texte Chapitre 2.3. Cette représentation intermédiaire, propre au modelleur, sert de support au calcul de la représentation *explicite* par le modelleur. Les capacités de traduction et de résolution de la représentation *symbolique* dépendent des mécanismes du modelleur. Ces capacités définissent ce que peut représenter le concepteur et la façon dont il peut le faire (doit-il tout décrire ? dans quel ordre ? que peut-il laisser en suspens ?).

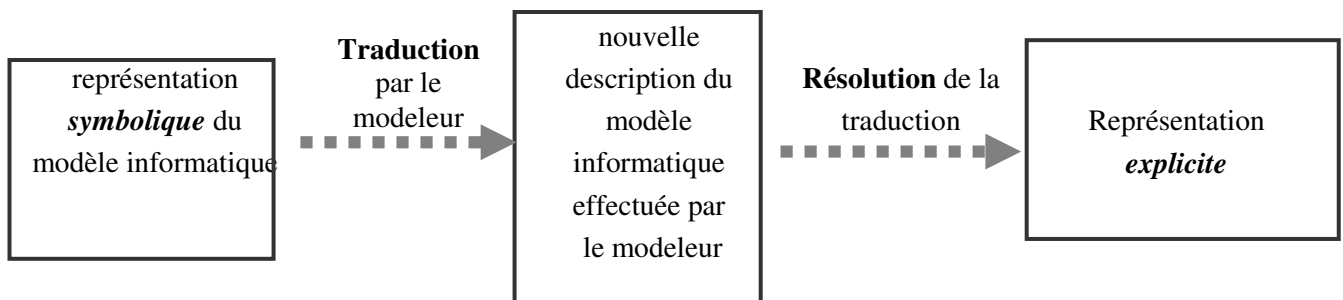


Figure 6: Passage d'une représentation *symbolique* à une représentation *explicite*

Pour aborder ces mécanismes, nous nous appuyons ici sur la catégorisation générale proposée par Agbodan (Agbodan 2002), proche de la catégorisation proposée par Maculet et Daniel (Maculet & Daniel 2005). Agbodan, comme Maculet et Daniel, distingue deux approches dans les mécanismes des modelleurs paramétriques : l'approche fonctionnelle, dite aussi paramétrique ou constructive, et l'approche équationnelle, dite aussi variationnelle.

La description de ces deux approches sera complétée par l'explicitation du mécanisme « par système de propagation » (R. Aish & Woodbury 2005; Woodbury 2010). Ce mécanisme est un cas particulier de l'approche paramétrique fonctionnelle. Il est particulièrement présent dans les modelleurs paramétriques utilisés en architecture.

Avant d'aborder ces mécanismes, certains termes ayant de multiples acceptions, nous commencerons par expliciter les termes ambigus que sont : « variable » (ou donnée dépendante), « paramètre » (ou donnée indépendante) et « contrainte ». Nous définirons également l'opposition entre « contraintes orientées » et « contraintes non-orientées » qui nous permettra de situer la notion de « résolution de contraintes ».

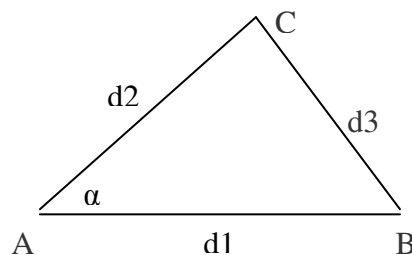
Au sein d'un programme informatique, des données peuvent être dépendantes d'un mécanisme et des résolutions précédentes : ce sont alors des données « dépendantes » (Woodbury 2010, p.14) ou « volatiles » (« Volatile data ») (Payne & Issa 2009, p.11). Les données volatiles ou dépendantes sont des « variables » au sens informatique du terme, puisqu'une variable est une « position de mémoire représentée par un identificateur et dont le contenu est censé varier au cours de l'exécution d'un programme » (Morvan et al. 2000, p.258).

Les données d'un modèle peuvent être assignées par le concepteur : elles sont alors « indépendantes » (Woodbury 2010, p.14) ou « persistantes » (Payne & Issa 2009, p.11). Les « données indépendantes » ou « persistantes » sont des « paramètres » au sens informatique du terme : en effet un paramètre est une « *variable dont le type est connu mais dont la valeur, l'adresse ou le nom ne seront précisés qu'à l'exécution* » (Morvan et al. 2000, p.179).

D'après Robert Woodbury, une contrainte est une formule comprenant des objets (c'est-à-dire des valeurs et des propriétés), des fonctions, des appels et des opérateurs (Woodbury 2010, p.14). Si le problème est correctement établi (c'est-à-dire ni sur-contraint ou sous-contraint), une fois évalué, il a pour résultat une valeur.

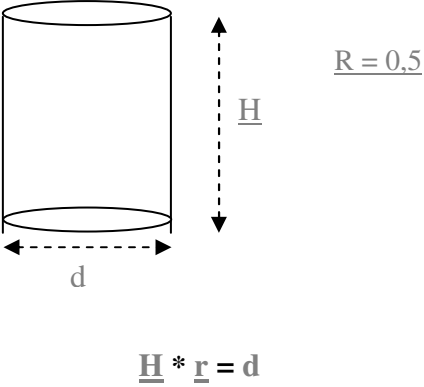
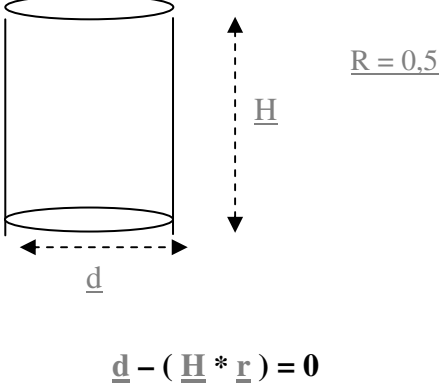
Par exemple, si on reprend le cas proposé par Maculet et Daniel (2005), d'un triangle défini par deux cotés (d1 et d2) et un angle ( $\alpha$ ) (cf. Figure 7), on obtient :

- les « variables indépendantes », c'est-à-dire les paramètres : d1, d2 et  $\alpha$  ;
- les « variables dépendantes » : d3 et les positions des points A, B et C ;
- des contraintes de coïncidence entre les points (le triangle doit rester fermé), les contraintes de longueurs de d1 et d2, ainsi que la contrainte de l'angle  $\alpha$ .



**Figure 7: Exemple d'un triangle contraint par des paramètres de longueur d1 et d2 ainsi qu'un paramètre d'angle  $\alpha$**

La résolution de contrainte est un mécanisme qui peut prendre deux formes différentes : 1-celle d'une fonction explicite, dans ce cas là elle est dite « orientée » (Maculet & Daniel 2005, p.8) ou « fonctionnelle » (Agbodan 2002, p.34); ou bien 2-celle d'une fonction algébrique, elle est alors dite « non orientée » (Maculet & Daniel 2005, p.7) ou « équationnelle » (Agbodan 2002, p.34). Si l'on prend l'exemple d'un cylindre dont le diamètre  $d$  dépend de la hauteur  $H$  selon un rapport  $r$  (cf. Tableau 3), alors la contrainte entre ces variables peut s'exprimer de ces deux façons : sous la forme «  $H*r = d$  » ou «  $d-(H*r) = 0$  ». La contrainte qui prend la forme «  $d-(H*r) = 0$  » est non-orienté car les trois variables, diamètre, hauteur et rapport, peuvent être changées (cf. Tableau 3 cas 2). La contrainte qui prend la forme «  $d=H*r$  » est orientée (cf. Tableau 3 cas 1), elle sous-entend la séquence de résolution de contraintes suivante : dans un premier temps, définir  $H$  et  $r$  et, dans un deuxième temps calculer  $d$ .

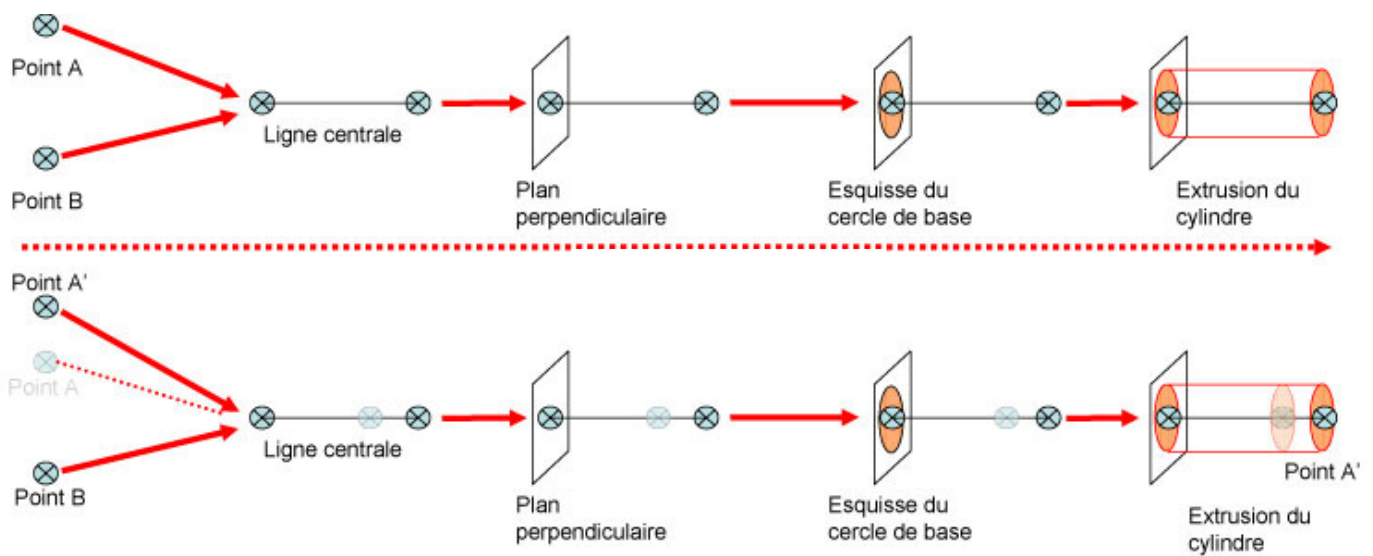
|   |  |
|---|--|
|  <p style="text-align: center;"><math>\underline{H} * \underline{r} = \underline{d}</math></p> |  <p style="text-align: center;"><math>\underline{d} - (\underline{H} * \underline{r}) = \underline{0}</math></p> |
| <p>Contraintes « orientées » : seuls le rapport <math>r</math> et la hauteur <math>H</math> sont modifiables</p>  | <p>Contraintes « non-orientées » : le rapport <math>r</math>, la hauteur <math>H</math> et le diamètre <math>d</math> sont modifiables</p>   |

**Tableau 3: Exemple du cylindre: contraintes « orientées » ou « fonctionnelles » Versus contraintes « non orientées » ou « équationnelles »**

### 2.1.2.1 L'approche dite «paramétrique fonctionnelle »

Deux approches principales sont utilisées pour les mécanismes des modeleurs paramétriques : l'approche paramétrique fonctionnelle et l'approche paramétrique variationnelle.

L'approche paramétrique fonctionnelle (Agbodan 2002, p.36) est aussi nommée approche « paramétrique » par Maculet et Daniel (Maculet & Daniel 2005, p.8). Cette approche s'appuie sur une représentation du modèle informatique au moyen d'un « schéma de construction » (Mineur 2009). Ce schéma de construction est constitué de paramètres et de contraintes, définies par la séquence des opérations de construction géométrique effectuées pour obtenir l'objet final. Par exemple, dans le cas de la construction d'un cylindre décrit par la Figure 8, deux variables indépendantes « Point A » et « Point B » sont définies. A partir de ces paramètres, un segment est construit puis, un plan normal au segment. Un cercle de base est construit sur le plan (le rayon est également une variable indépendante) et utilisé pour extruder un cylindre le long du segment. Dans ce cas, la définition des variables indépendantes (les deux points et le rayon), des variables dépendantes (le segment, le plan, le cercle et le cylindre) ainsi que des contraintes (de coïncidence, de perpendicularité et d'égalité de longueur) résulte de l'historique de conception.

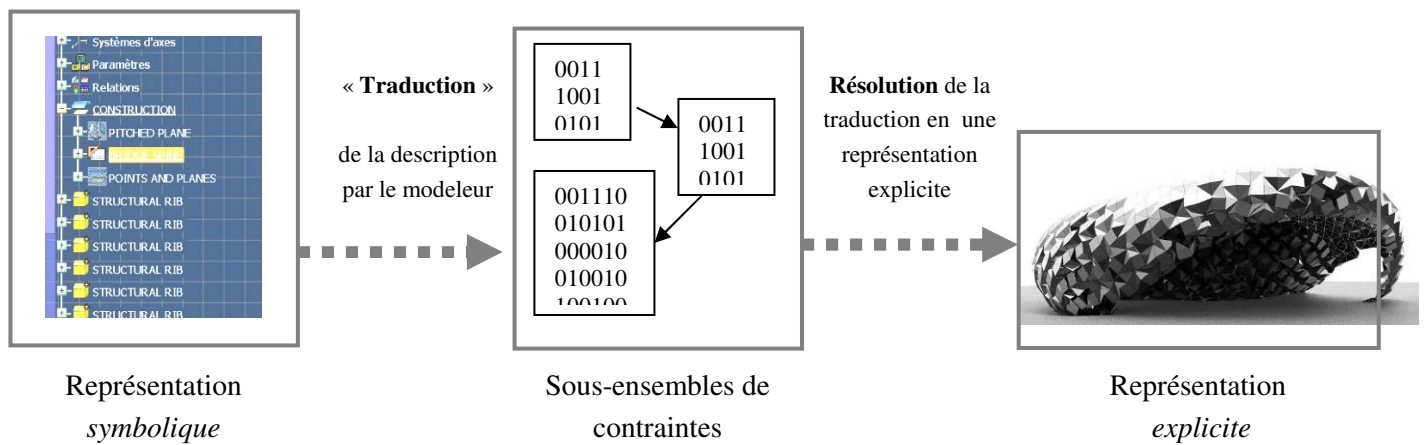


**Figure 8: Exemple de mise en œuvre de l'historique de construction pour la résolution de contrainte d'un cylindre avec le modelleur CATIA, Extrait de la formation à Digital Project prodiguée par Gehry Technologies Europ, Source : <http://www.gtwiki.org> (consulté le 20 avril 2011)**

Les étapes de construction du modèle représentées dans un schéma de construction sont jouées chronologiquement en vue d'obtenir une représentation *explicite* du modèle. Dans l'exemple précédent, (cf. Figure 8) quand la variable indépendante « Point A » est modifiée en « Point A' », les séquences de construction sont rejouées avec le nouveau paramètre : la géométrie finale du cylindre est alors mise à jour.

L'exemple de la Figure 8 montre un cas de modèle relativement simple. Pour les modèles où les chaînes de dépendances entre variables sont plus complexes, ce schéma de construction peut être décomposé par le modelleur en sous-ensembles de contraintes, solvables les uns après les autres (cf. Figure 9). Un modelleur relevant d'une telle approche « paramétrique fonctionnelle », aborde un schéma de construction comme un problème composé de sous-problèmes pouvant être résolus de manière séquentielle (Lesage 2006)<sup>5</sup>.

<sup>5</sup> Pour David Lesage (Lesage 2006), un modèle paramétrique est : « un algorithme qui traite chacune des étapes de la construction d'un modèle géométrique comme un problème paramétrique », c'est-à-dire comme un problème « pouvant être décomposé en un ensemble de sous problèmes pouvant être résolus les uns après les autres, de façon séquentielle » (Lesage 2006).



**Figure 9: Passage d’une représentation symbolique à une représentation explicite dans le cas de l’approche paramétrique fonctionnelle**

Nous venons de le voir, l’approche « paramétrique fonctionnelle » correspond à une résolution de contraintes séquentielle et constructive (Maculet & Daniel 2005, p.8). Séquentielle car les contraintes sont organisées en sous-ensembles. Et constructive car l’ordre de résolution de ces sous-ensembles est défini par la construction du modèle.

Cela implique que les contraintes doivent être « orientées » (Maculet & Daniel 2005, p.8) (cf. paragraphe précédent) : les contraintes ont un ordre de succession, une contrainte ne peut pas être résolue avant l’autre si cela ne correspond pas au schéma de construction. Par exemple dans la Figure 8, on ne peut pas définir le cylindre avant d’avoir construit la ligne entre les points A et B : cette ligne est nécessaire à la construction.

Ces caractéristiques font de l’approche « paramétrique fonctionnelle » une méthode dont le calcul des modèles est rapide et utilisable pour des géométries en trois dimensions. Cette approche est la plus courante au sein des logiciels utilisés en conception architecturale, en particulier sous la forme plus restrictive des « systèmes basés sur la propagation », que nous décrivons plus précisément dans le paragraphe 2.1.3.

Malgré ses avantages, les mécanismes « paramétriques fonctionnels » ont un inconvénient : les problèmes géométriques doivent y être bien contraints. Un modèle sur-contraint ou sous-contraint ne peut pas être résolu par un tel mécanisme.

### 2.1.2.2 L'approche dite « paramétrique variationnelle »

Contrairement à l'approche « paramétrique fonctionnelle », l'approche « paramétrique variationnelle » permet de traiter des paramètres et des contraintes indépendamment des opérations de construction géométrique et de leurs chronologies. C'est une approche globale des contraintes, opérant par un système d'équations (Maculet & Daniel 2005, p.9). Cette approche variationnelle est parfois dite déclarative ou équationnelle.

Contrairement au paramétrique fonctionnel, les contraintes y sont « non-orientées » (cf. Tableau 3), l'approche paramétrique fonctionnelle quant à elle ne peut résoudre que des contraintes orientées (cf. 2.1.2.1).

En mathématiques, un problème variationnel est un problème pouvant être décomposé en sous-problèmes solvables simultanément (alors que les sous-problèmes fonctionnels ne sont solvables que de façon séquentielle). Un système équationnel met en œuvre une décomposition en sous-problèmes qui peut être reconçue à tout moment. Ce mécanisme permet une approche globale des contraintes. Néanmoins, il nécessite un temps de calcul important. Cette approche n'est donc utilisée que pour des constructions géométriques en deux dimensions, dans des fonctionnalités de modeleurs appelées « sketchers » (présents sur les logiciels CATIA ou TopSolid par exemple).

### 2.1.3 Les systèmes basés sur la propagation (« *Propagation Based Systems* »)

Les mécanismes « basés sur la propagation » tels que décrit par Aish et Woodbury dans « Multi-Level interaction in Parametric Design » (2005) et Woodbury dans son ouvrage *Elements of Parametric Design* (2011), concernent le fonctionnement de la majorité des modeleurs paramétriques utilisés en architecture.

Aish et Woodbury situent les mécanismes « basés sur la propagation » par rapport aux mécanismes « graph based » distingués par Hoffmann et Joan-Arinyo dans *A Brief on Constraint Solving* (Hoffman & Joan-Arinyo 2005). Ces auteurs distinguent trois approches : « graph-based », « logic-based » ou « algebraic », que nous exposons rapidement ci-dessous. La résolution de problème « graph-based » peut être rapprochée de l'approche « paramétrique fonctionnelle » vue précédemment. Cette approche propose une traduction du modèle numérique en graphe où les nœuds représentent les objets géométriques et les arcs (les relations entre les nœuds) représentent les contraintes qu'ils entretiennent. Ces graphes sont ensuite parcourus pour être résolus (Hoffman & Joan-Arinyo 2005, p.8)<sup>6</sup>. Dans ce graphe, des sous-problèmes facilement solvables sont identifiés (Woodbury 2010, p.12). Ces sous-ensembles composent des séquences d'étapes de construction<sup>7</sup> associables et permettant de générer la géométrie du modèle. La résolution de contraintes « logic-based », quant à elle, traduit le problème en axiomes, solvables par l'inférence de règles logiques (Hoffman & Joan-Arinyo 2005, p.12)<sup>8</sup>. Enfin, la

---

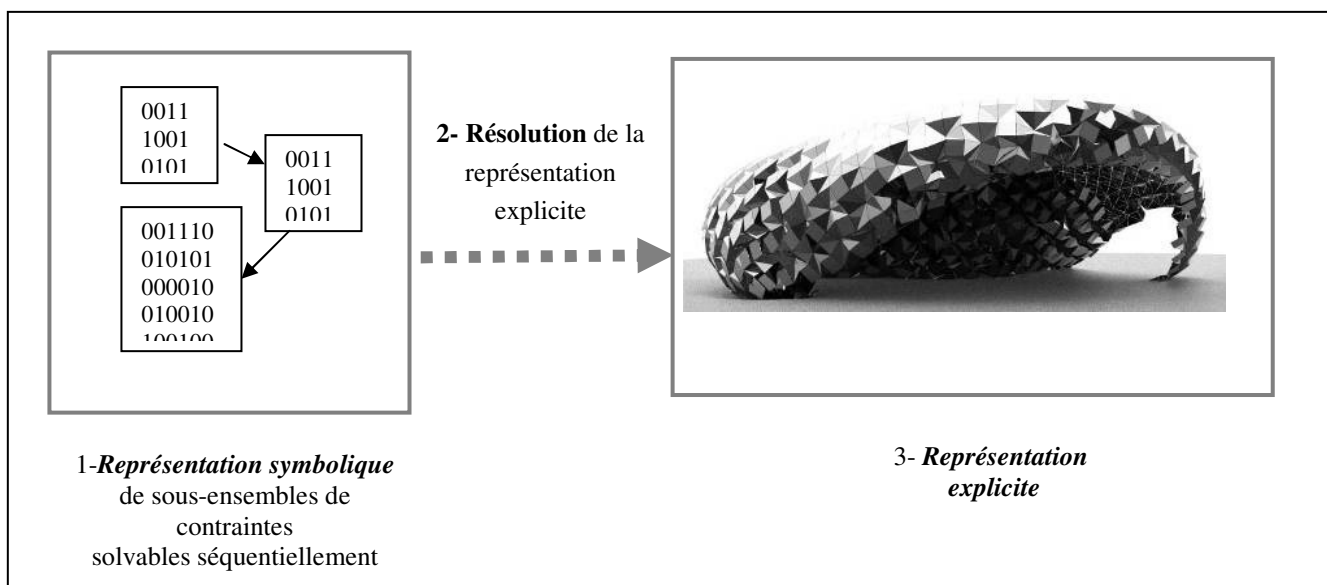
<sup>6</sup> « *The solver analyzes the graph and formulates a solution strategy whereby subproblems are isolated and their solutions suitably combined* » (Hoffman & Joan-Arinyo 2005, p.8).

<sup>7</sup> « *symbolic sequence of basic construction steps* » (Hoffman & Joan-Arinyo 2005, p.8).

<sup>8</sup> « *the problem is translated into a set of assertions and axioms characterizing the constraints and the geometric objects. By employing reasoning steps, the assertions are transformed in ways that expose solution steps in a*

résolution de contrainte « algebraic » traduit le problème en un système d'équations non linéaires, ensuite résolues par des méthodes de calculs spécifiques à ces équations (Hoffman & Joan-Arinyo 2005, p.13). Cette dernière approche peut être assimilée à l'approche « paramétrique variationnelle » décrite précédemment (cf. 2.1.2.2).

Les systèmes basés sur la propagation sont un cas particulier de l'approche paramétrique fonctionnelle. Ils en reprennent les principales caractéristiques : ce sont des systèmes séquentiels et constructifs, mais supposent que le concepteur décrive une représentation *symbolique* qui puisse être directement résolue par le modelleur (Woodbury 2010, p.12). Là où les modelleurs reprenant de l'approche paramétrique fonctionnelle traduisent un modèle en sous-ensembles de problèmes solvables en un ordre précis, les modelleurs basés sur la propagation n'opèrent pas cette traduction. Les systèmes basés sur la propagation nécessitent que le dessinateur décrive directement la représentation symbolique sous la forme de graphes (cf. Figure 10-1) pouvant être résolus (cf. Figure 10-2) par le modelleur en représentations explicites (cf. Figure 10-3). A partir de la représentation symbolique, le modelleur propage les valeurs des variables, en partant des données indépendantes connues pour calculer les variables dépendantes, en parcourant le graphe orienté.



**Figure 10: Passage d'une représentation symbolique à une représentation explicite dans le cas d'un système basé sur la propagation**

Woodbury décrit les deux algorithmes sollicités par un modelleur basé sur un mécanisme de propagation comme suit :

- **Un algorithme de mise en ordre** : cet algorithme définit un ordre dans les nœuds, de façon à ce que chaque nœud ne soit parcouru qu'une seule fois, et seulement lorsque tous ses prédécesseurs ont déjà été évalués. Plusieurs ordres de parcours sont possibles et également valables. Ces ordres sont nommés topologiques (Woodbury 2010, p.15).

---

*stereotypical way and special solvers then compute coordinate assignments.* » (Hoffman & Joan-Arinyo 2005, p.12).



- **Un algorithme de propagation** : cet algorithme évalue chacun des nœuds selon l'ordre défini précédemment. Cet algorithme a pour résultat le calcul d'une représentation *explicite*. Cette représentation explicite est l'instance du modèle paramétrique (Woodbury 2010, p.15).

Nous pouvons illustrer ce fonctionnement en prenant pour exemple la modélisation d'un cylindre sur le plugin Grasshopper, un modèleur basé sur la propagation. Pour cet exemple nous reprenons les contraintes et les variables exposées dans la Figure 8 et décrit dans le paragraphe 2.1.2.1. Cette modélisation est illustrée dans la Figure 11. On observe que la représentation symbolique du modèle paramétrique reprend la description par historique de construction donné précédemment en exemple (cf. Figure 11-B).

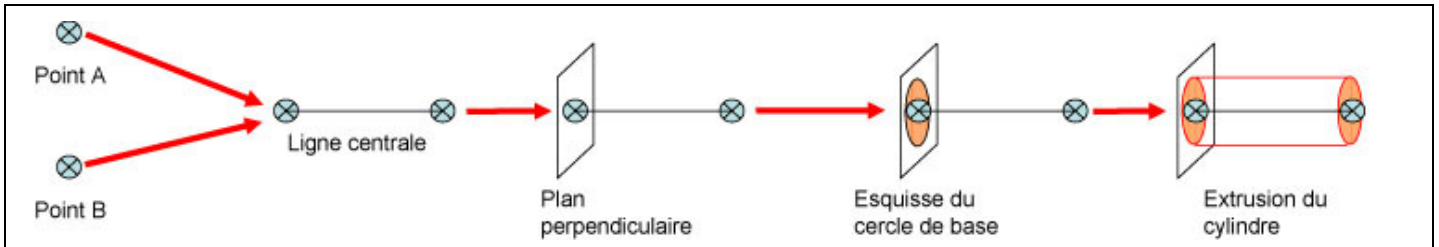


Figure 11A: description de la géométrie représentée: les flèches indiquent un ordre chronologique lié à l'historique de construction

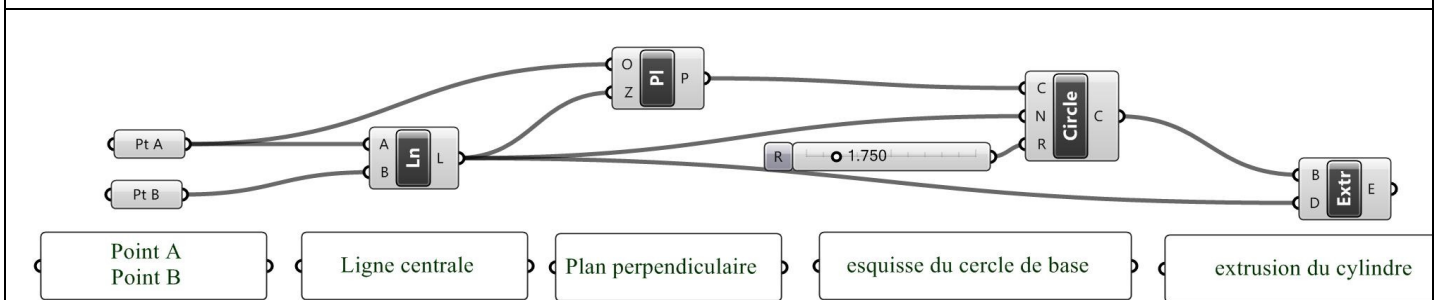


Figure 11 B : Proposition de modélisation de l'exemple sur Grasshopper, les données sont propagées de la gauche vers la droite

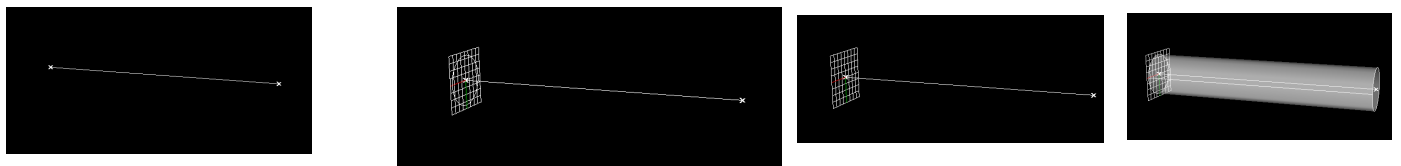


Figure 11C : Illustration de la résolution du modèle au fur et à mesure de sa construction

**Figure 11 : Exemple de la modélisation d'un cylindre au moyen d'un système basé sur la propagation**

Dans un système basé sur la propagation, un modèle paramétrique prend la forme d'un graphe (Woodbury 2010). Woodbury décrit un graphe comme un ensemble de nœuds connectés par des liens (Woodbury 2010, p.13), aussi nommés arcs. Dans le cas de la modélisation paramétrique, chaque nœud est un « schéma » possédant un nom et des propriétés. Dans le cas d'un nœud représentant un point (cf. Figure 12), le nœud peut comprendre : le nom du nœud, par exemple « Point p », les caractéristiques du point : dans cet exemple, ce point se situe dans le système de coordonnées « cs » et possède comme coordonnées  $x=3$ ,  $Y=4$  et  $Z=1$ .

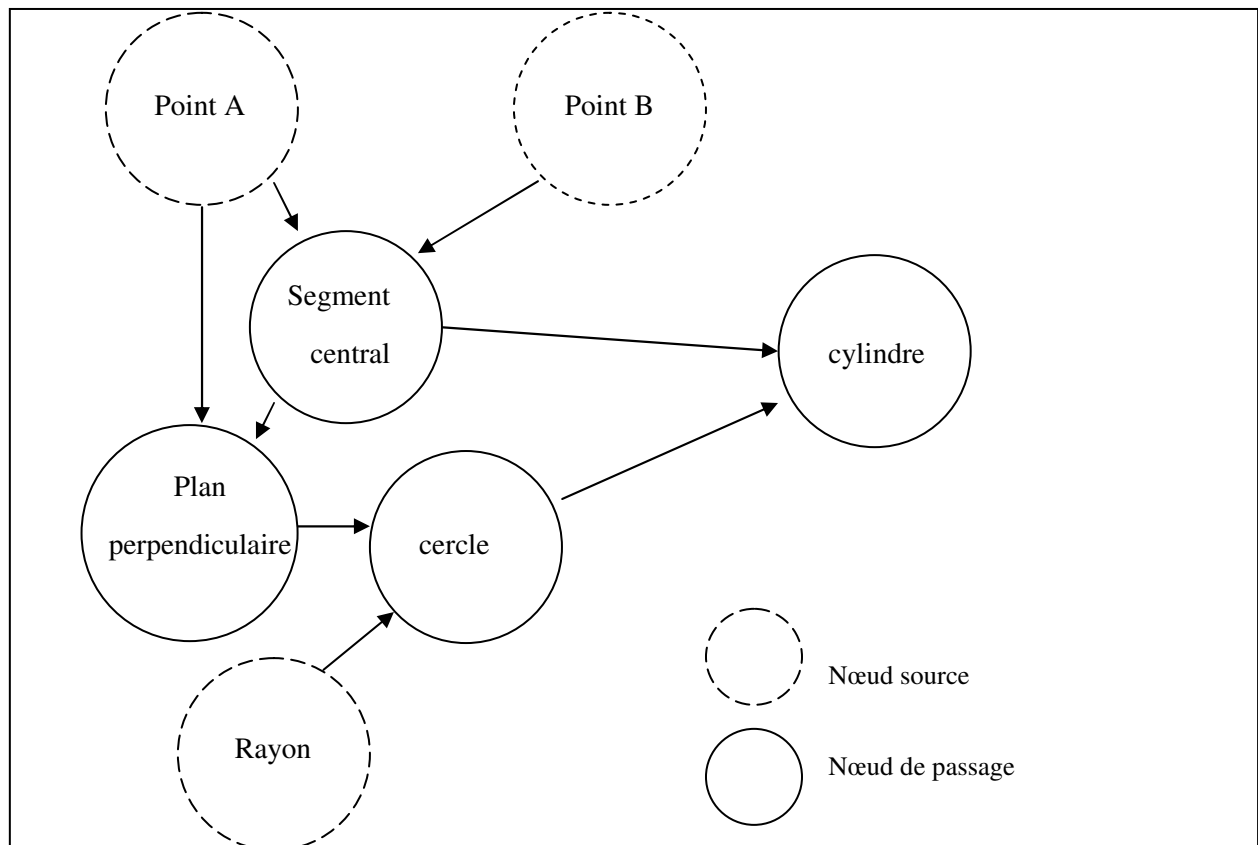
```

Point p  {
    CoordSystem : cs ;
    X : 3.0 ;
    Y : 4.0 ;
    Z : 1.0 ;
}

```

**Figure 12 : Exemple d'un nœud "point p" (Woodbury 2010, p.13)**

Les nœuds sont liés entre eux par des contraintes. Dans l'acception proposée par Woodbury du terme contrainte, celui-ci désigne de nombreuses sortes de relations. Pour Woodbury, une contrainte est une formule comprenant des objets, des fonctions et/ou des opérateurs, et retournant des valeurs. Deux nœuds sont liés par une contrainte quand un nœud successeur a besoin d'une valeur de son prédécesseur pour définir ses propriétés. Par exemple, dans le cas précédent du cylindre (cf. Figure 4), la modélisation par Grasshopper peut être traduite en un graphe (cf. Figure 13), où par exemple le nœud « segment principal » est lié aux nœuds « Point A » et « Point B » par des contraintes d'incidence.

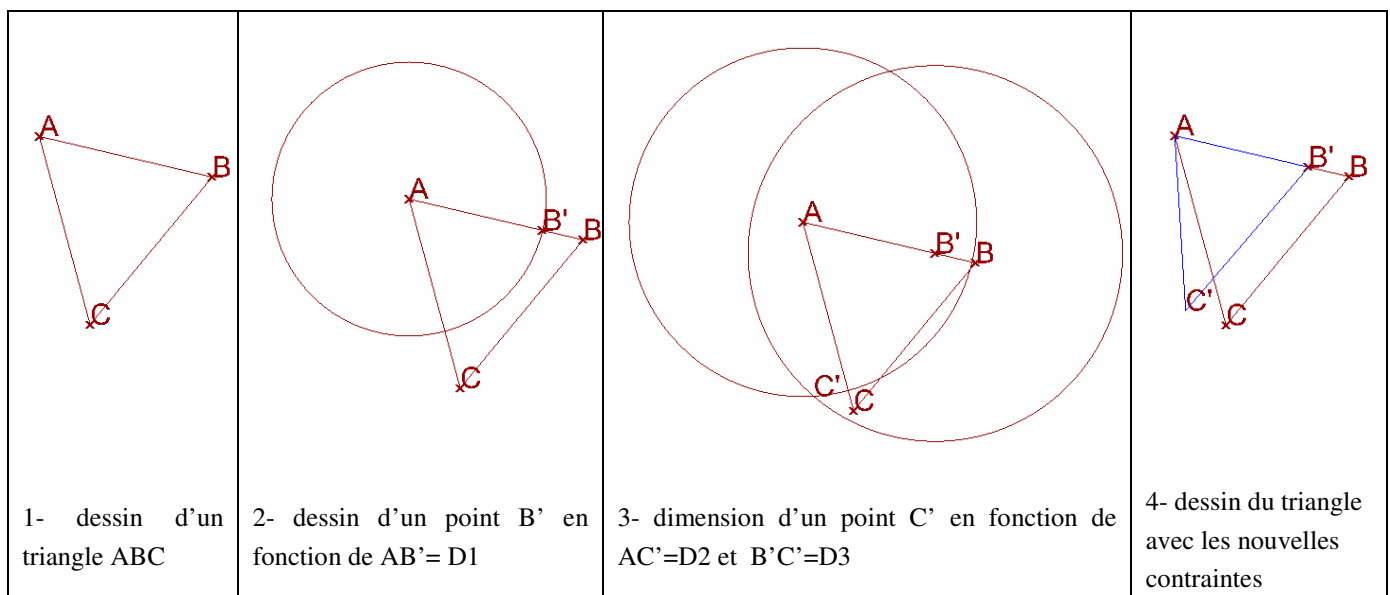


**Figure 13 : Graphe correspondant à l'exemple du cylindre, exposé dans la Figure 4**

Tous les nœuds ne sont pas équivalents. Des nœuds peuvent n'avoir que des propriétés assignées, ou persistantes (c'est-à-dire indépendantes du graphe), ce sont alors des « nœuds sources », que l'on nomme aussi paramètres (Woodbury 2010, p.14). Des nœuds peuvent contenir des propriétés volatiles, dépendantes du graphe, ce sont alors des nœuds de passage (« sink nodes ») (Woodbury 2010, p.14).

Le système évalue l'ordre dans lequel il peut résoudre le graphe. Il ne peut pas évaluer un nœud plusieurs fois : sinon cela signifierait qu'un nœud doit être calculé pour être calculé, ce qui est un paradoxe que le système de propagation ne peut résoudre. Cela s'observe par exemple si l'on tente de modéliser le cas du triangle donné par Maculet et Daniel (2005). Ces auteurs donnent l'exemple d'un triangle ABC, quelconque, dont on contraint *a posteriori* les trois dimensions des côtés. Si l'on modélise ce problème sous forme de graphe, on se rend compte qu'il n'est pas directement modélisable. En effet, il nécessite que les nœuds Point A, Point B et Point C soient évalués deux fois : une première fois pour construire le premier triangle. Puis, lors de la nouvelle contrainte pour les « mettre à jour » avec les nouvelles distances.

Si l'on veut représenter ce problème avec un mécanisme de propagation, de nouveaux nœuds doivent être créés pour les points que les nouvelles distances modifient. Cela s'observe dans la Figure 14, rejouant la séquence géométrique de ce système de contraintes, basé sur la propagation: des points B' et C' doivent être créés.



**Figure 14 : Construction de la représentation explicite de l'exemple du triangle dans le cas d'un mécanisme basé sur la propagation**

L'exemple de la Figure 14 illustre que dans ce système, comme pour toute approche paramétrique fonctionnelle, les contraintes sont orientées (*cf.* 2.1.2.1). C'est pourquoi un modèle paramétrique constitué grâce à un mécanisme basé sur la propagation est dit « dirigé ». Un modèle paramétrique constitué grâce à un mécanisme basé sur la propagation est également dit « acyclique », car il est résolu de façon séquentielle, un même nœud ne peut pas être parcouru plusieurs fois.

Pour Woodbury, le principal avantage des systèmes paramétriques basés sur la propagation est la simplicité d'usage (Woodbury 2011 p. 12). La construction de représentations propres à être résolues permet au concepteur d'explicitier exactement les variables et les contraintes de son modèle ainsi que leur résolution. Cela permet au concepteur un contrôle absolu sur la génération de sa géométrie. De plus, cette simplicité permet au modèleur des résolutions rapides et efficaces, et donc, la représentation de géométries complexes.

Néanmoins, cette simplicité de représentation demande des compétences d'usages des modeleurs et de programmation. Par ailleurs, cette simplicité induit un certain nombre de limites techniques. Ainsi certaines géométries qui nécessitent une représentation cyclique, comme les géométries fractales (qui nécessite de réutiliser des géométries produites) (de Boissieu & Guéna 2012) ou encore les structures en tenségrité (Woodbury 2010, p.12) ne peuvent être représentées avec un modeleur paramétrique basé sur la propagation.

La visualisation et la compréhension du flux de données dans la représentation *symbolique* du modèle est, pour Woodbury, indispensable à l'usage des modeleurs paramétriques faisant appel à ces mécanismes. Les flux de données peuvent prendre des structures complexes. Dans certains cas, les nœuds peuvent contenir plusieurs propriétés, ce sont alors des nœuds multiples, considérés comme l'association de plusieurs nœuds simples (Woodbury 2010, p.17; R. Aish & Woodbury 2005, p.6). Dans ce cas, des traitements semblables aux boucles de programmation peuvent être mis en œuvre.

La Figure 15 et la Figure 16 donnent l'exemple de deux modèles utilisant des nœuds multiples. Dans la Figure 15, on peut voir un nœud Point, relié à deux nœuds contenant chacun les valeurs multiples 0, 1, 2, 3 et 4. Ces nœuds multiples de valeurs permettent de générer cinq points: le point 1 possède les coordonnées (0, 0, 0), le point 2 (1, 1, 0), le point 3 (2, 2, 0) *etc.* Dans la Figure 16, les mêmes nœuds « point » et « valeurs » sont utilisés. Un nouveau nœud a cependant été introduit pour transformer la structure des données. Ce nœud génère le produit cartésien des valeurs des nœuds multiples, produisant toutes les combinaisons possibles entre les valeurs : (0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 3, 0), (0, 4, 0), (1, 0, 0), *etc.* Cette manipulation des données, permises par les nœuds multiples, permet de complexifier le modèle.

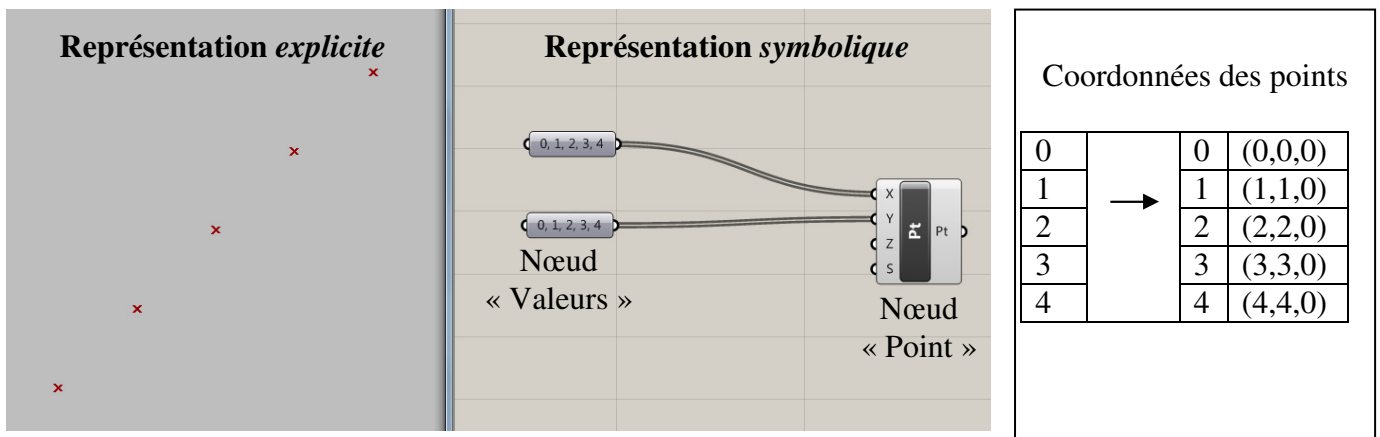


Figure 15 : Exemple de nœuds multiples

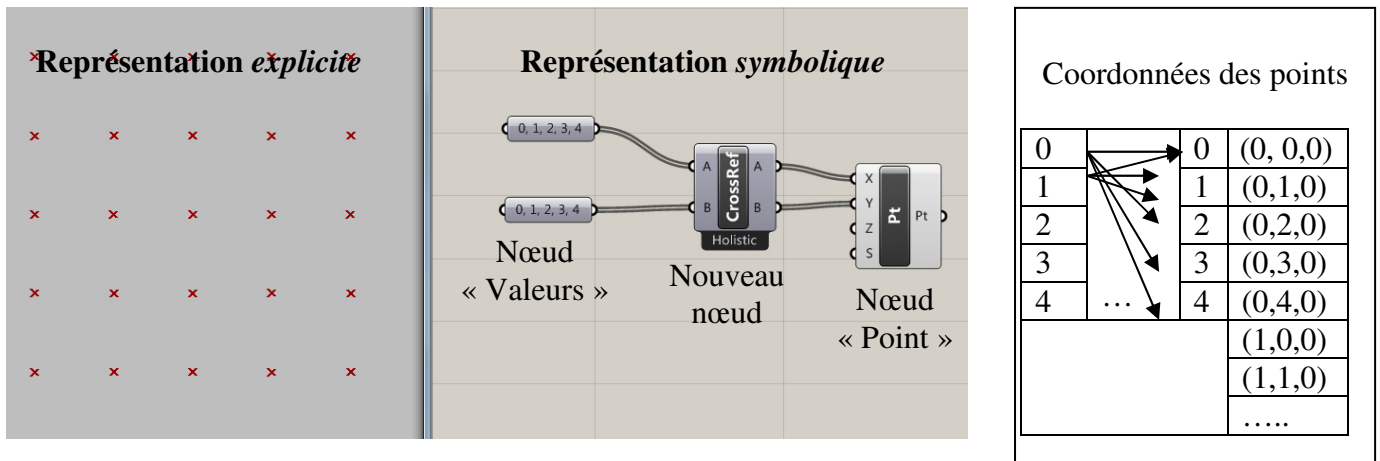


Figure 16 : Exemple de nœud multiple avec transformation de la structure des données

Si les nœuds qui contiennent plusieurs propriétés sont intéressants, ils sont également difficiles d'utilisation : l'organisation des données peut devenir peu lisible et poser des questions d'interface (Woodbury 2012 p.19). Le problème qui se pose porte sur la structuration de ces données au sein de ces nœuds aux valeurs multiples.

Les mécanismes que nous venons de définir, sont mis en œuvre dans des modeleurs utilisés en architecture pour la conception. Comme nous nous intéressons spécifiquement à l'usage de modeleurs dits paramétriques pour la conception architecturale, nous proposons, dans la suite de ce chapitre, d'interroger ces modeleurs. Ces modeleurs seront ici abordés grâce aux spécificités des mécanismes paramétriques que nous venons d'identifier.

## 2.2 Mises en œuvre de mécanismes paramétriques : caractéristiques de modelleurs

### 2.2.1 Origines et évolution des modelleurs paramétriques

Les tous premiers modelleurs procèdent de la recherche militaire (Picon 2010, p.28), puis de la recherche industrielle (Weisberg 2008). C'est toutefois à l'université, dès 1960, que la question de la résolution de contraintes pour la modélisation géométrique est abordée dans la recherche « Sketchpad » développée par Ivan Sutherland (Sutherland 1963). Sketchpad est un modelleur proposant un algorithme de propagation de valeurs ainsi qu'un solveur de contraintes. Un opérateur peut intervenir sur une représentation géométrique 2D au moyen d'un stylet et de boutons poussoirs activant des contraintes (parallélisme, perpendicularité, incidence, *etc.*) (*cf.* Figure 17). Les contraintes du modèle peuvent être définies *a posteriori*.



Figure 17 : "TX-2 Operating area-sketchpad in use" illustration 1.2 de *Sketchpad : A man-machine graphical communication system* (Sutherland 1963)

Différents modelleurs sont développés par la suite, malgré les difficultés posées par les capacités réduites de visualisation et de calcul du matériel informatique de l'époque. Ces modelleurs sont en particulier développés comme des logiciels métiers pour l'aéronautique et la mécanique. Par exemple, en 1970, la société Matra Datavision développe en France le modelleur paramétrique « Euclid » et Dassault System développe le modelleur CATIA (Weisberg 2008).

### 2.2.2 Description de quelques uns des modelleurs paramétriques utilisés pour l'architecture

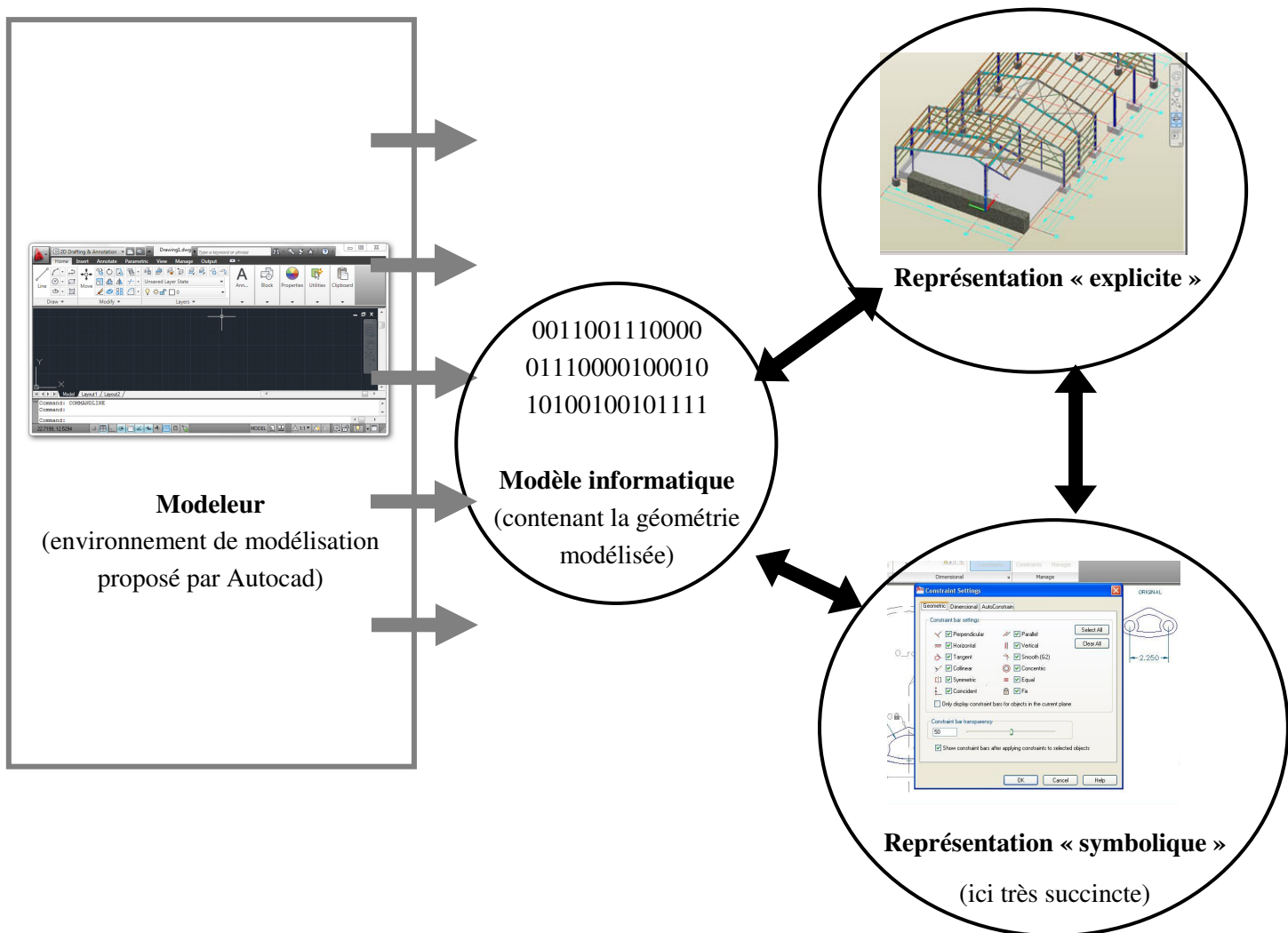
Les modelleurs que nous interrogeons ici répondent à trois critères :

- 1- ils sont largement utilisés en agence d'architecture ;

2- ils permettent à l'utilisateur d'intervenir sur un modèle numérique au travers de représentations *explicites* ainsi que de représentations *symboliques* et,

3- ils permettent la mise en œuvre de mécanismes dits paramétriques (paramétrique variationnel, fonctionnel ou basé sur la propagation) tels que définis dans la section 2.1.

Nous avons volontairement retenu une définition large de la modélisation paramétrique, en vue d'interroger des modeleurs, utilisés en architecture, et mettant en œuvre des mécanismes paramétriques très différents. Par exemple, certains mécanismes intégrés à Autocad depuis 2010 nous permettent de l'intégrer aux modeleurs interrogés ici (cf. Figure 18). Cependant, nous verrons dans la suite du texte que, dans la grande variété des mises en œuvre de ces mécanismes, seul un nombre restreint de celles-ci sont intéressantes à retenir pour le corpus de la présente recherche.



**Figure 18: Modeleur, représentation *explicite* et représentation *symbolique*, le cas d'Autocad 2010**

Nous aborderons ici trois sortes de modeleurs comprenant des mécanismes paramétriques : les modeleurs généralistes utilisés dans différents secteurs dont l'architecture et les modeleurs métiers pour

l'architecture en distinguant ceux développés autour de logiciels généralistes et ceux conçus indépendamment d'autres modeleurs.

#### - Les modeleurs métiers

Au sein des modeleurs développés spécifiquement pour la pratique de l'architecture, certains intègrent des mécanismes paramétriques (cf. Tableau 4). Ces outils sont, pour la plupart, des modeleurs « paramétriques à base d'objets » (Eastman et al. 2008): ils proposent des objets prédéfinis (fenêtres, murs, escaliers, etc.) dont les variables indépendantes (les éléments modifiables dans ces objets) et les contraintes (les relations entre les objets, leurs positions possibles, etc.) sont déterminées par l'éditeur du logiciel. L'utilisateur a cependant la possibilité de créer d'autres familles d'objets mais cela nécessite des compétences dont tous les utilisateurs ne disposent pas nécessairement. Les objets prédéfinis permettent un usage relativement facile du modeleur (par exemple il n'est pas nécessaire de construire une fenêtre, elle est déjà conçue) mais aussi très contraint (je ne peux pas positionner une fenêtre en dehors d'un mur, par exemple). Ces modeleurs sont le plus souvent orientés vers une utilisation BIM (Building Information Modelling) (Eastman et al. 2008) pour permettre l'interopérabilité avec les logiciels des différents intervenants d'un projet d'architecture (comme les outils de calcul et d'évaluation des BET par exemple) et le travail à plusieurs sur un même modèle. Les modeleurs de ce type les plus connus, sont Revit, ArchiCad, Vectorworks ou AllPlan. ParaCloud et VasariProject sont également des modeleurs paramétriques métiers, développés à base d'objets spécifique à l'architecture. Néanmoins ils ne sont encore que peu utilisés.

Les modeleurs paramétriques à base d'objets facilitent la modélisation mais la contraignent également en nécessitant l'utilisation d'objets prédéfinis ou de nouvelles familles d'objets. Ce manque de liberté de modélisation fait que nous ne les avons pas retenus pour le corpus, même si certains sont très couramment utilisés en agence d'architecture.

| <i>Modeleurs</i>                       | <i>Méthodes utilisées</i>               | <i>Remarques</i>   | <i>Domaine d'application originel</i> |
|--|---|--|---------------------------------------|
| <b>Modeleurs paramétriques métiers</b> |   |  |                                       |
| <b>ArchiCAD</b><br>(graphisoft)        | par features<br>(méthode fonctionnelle) | L'existence d'objets modifiables permet une modélisation paramétrique fonctionnelle réduite, basée sur l'utilisation de features.  | Architecture                          |
| <b>Revit</b><br>(Autodesk)             | par features<br>(méthode fonctionnelle) | Propose des objets comprenant des données indépendantes modifiables.   | Architecture                          |
| <b>Vasari Project</b><br>(Autodesk)    | par features<br>(méthode fonctionnelle) | Propose des objets comprenant des données indépendantes modifiables. Vasari project est un logiciel prévu pour modéliser des projets d'architecture ou d'urbanisme et pour en analyser | Architecture                          |



|                                |  | les performances énergétiques   |           |
|--------------------------------|--|---|-----------|
| <b>Vectorworks</b><br>(Cesyam) | par features<br>(méthode<br>fonctionnelle) | Propose des objets comprenant des données indépendantes modifiables. Vectorworks est quelques fois utilisé en architecture (cf. ceysam.fr). | Mécanique |

**Tableau 4 : Quelques exemples de modeleurs paramétriques métiers développés pour l'architecture**

**- Les modeleurs développés pour d'autres métiers et réappropriés en architecture**

De nombreux logiciels développés pour d'autres pratiques, en particulier la mécanique ou l'animation, ont été appropriés par des architectes pour concevoir (Estevez 2001; Lynn 1997) (cf. Tableau 5).

Les modeleurs développés à l'origine pour la mécanique ou l'aéronautique et aujourd'hui utilisés en architecture sont principalement AutoCad, TopSolid ou encore CATIA. AutoCad propose, depuis la version 2010, un module permettant de construire un modèle paramétrique variationnel en deux dimensions. TopSolid, quant à lui, est un modeleur développé à l'origine pour l'industrie automobile. Il propose un mécanisme paramétrique fonctionnel permettant de modéliser en trois dimensions, complété par un mécanisme paramétrique variationnel permettant de générer des esquisses en deux dimensions. Le fonctionnement de TopSolid est proche de celui de CATIA, qui sera étudié plus précisément dans le paragraphe 2.3.3.

Certains modeleurs développés pour l'animation proposent également une représentation *explicite* accompagnée d'une représentation *symbolique* succincte. C'est le cas de Maya et de 3DStudioMax, où des « piles de modificateurs » permettent de visualiser l'historique de transformation d'une forme et, facilitent ainsi la modélisation d'une forme en mouvement.

Les mécanismes paramétriques proposés par AutoCad, ou encore dans Maya et 3DStudioMax sont trop limités ou trop spécifiques à un usage pour être intégrés dans le corpus de notre recherche. L'usage de CATIA est interrogé dans cette recherche, au travers de sa version métier Digital Project, présentée dans le paragraphe suivant.

| <i>Modeleurs</i>   | <i>Méthodes utilisées</i>           | <i>Remarques</i>  | <i>Domaine d'application originel</i> |
|--|-------------------------------------|---|---------------------------------------|
| <b>Modeleurs paramétriques réappropriés par l'architecture</b> |                                     |   |                                       |
| <b>Autocad</b><br>(Autodesk)                                   | Variationnelles                     | des contraintes peuvent être établies entre les objets depuis Autocad2010.  | Dessin technique, mécanique           |
| <b>CATIA</b><br>(Dassault System)                              | Variationnelles et fonctionnelle    | <i>Cf. 2.3.3</i>  | Mécanique, aéronautique               |
| <b>TopSolid</b><br>(Missler Software)                          | Variationnelles et fonctionnelle    | TopSolid est un logiciel de modélisation proposant un mécanisme proche de celui de CATIA. Il n'est pas spécifique à l'architecture mais est utilisé par Bernard Cache (Cache 2010). | Mécanique                             |
| <b>Maya</b><br>(Autodesk)                                      | fonctionnelle<br>(par l'historique) | La lisibilité de l'historique des fonctions et la possibilité d'intervenir sur celui-ci permettent une certaine approche « fonctionnelle » réduite des contraintes.                 | Animation                             |
| <b>3DS Max</b><br>(Autodesk)                                   | fonctionnelle<br>(par l'historique) | La modélisation d'une succession de fonctions et la possibilité de leurs modifications permet une modélisation paramétrique fonctionnelle réduite.                                  | Animation                             |

**Tableau 5 : Quelques exemples de modeleurs paramétriques réappropriés par l'architecture**

- **Les outils métiers développés à partir de modeleurs généralistes ou spécifiques à d'autres pratiques**

Des logiciels, sous formes de plug-ins ou des extensions de fonctionnalités, ont été développés pour permettre à des architectes de développer des modèles paramétriques à partir de modeleurs généralistes (cf. Tableau 6). C'est le cas de Digital Project, version métier de CATIA pour les architectes, mais aussi de Grasshopper, plugin de Rhinocéros, de Design Script, extension d'Autocad ou encore de Generative Components, développé sur la base de MicroStation. Ils paraissent tous les quatre particulièrement intéressants du point de vue de leur usage en architecture et, ils seront décrits plus précisément dans la prochaine section du texte (cf. 2.3).

Le modèleur RhinoBIM, extension de Rhinocéros, propose, quant à lui, des fonctionnalités BIM ainsi que des objets paramétriques préconstruits spécifiques à l'architecture (poutres, portes, etc.). Mais comme pour le cas des modeleurs « paramétriques à base d'objets », les possibilités de conceptions de nouveaux objets et de nouvelles relations sont possibles (par la création de familles paramétriques), mais difficiles. Nous ne retenons donc pas RhinoBIM dans notre corpus.

| <i>Modeleurs</i>   | <i>Méthodes utilisées</i>            | <i>Remarques</i>                               | <i>Sur la base de :</i> |
|--|--------------------------------------|--|-------------------------|
| <b>Outils métiers pour l'architecture développés sur la base de modeleurs généralistes</b> |                                      |  |                         |
| <b>Digital Project</b><br>(Gehry Technologies, Dassault System)                            | Variationnelles et fonctionnelle     | <i>Cf. 2.3.3</i>                               | CATIA                   |
| <b>Rhino BIM</b><br>(McNeel)   | par features (méthode fonctionnelle) | Outil BIM, avec objets paramétriques préconçus | Rhino                   |
| <b>Grasshopper</b><br>(plugin de Rhinocéros3D, McNeel)                                     | fonctionnelle                        | <i>Cf. 2.3.1</i>                               | Rhino                   |
| <b>Design Script</b><br>(Version bêta) (Autodesk)  | fonctionnelle                        | <i>cf. 2.3.4</i>                               | Autocad                 |
| <b>Generative Components</b><br>(Bentley System)   | fonctionnelle                        | <i>Cf. 2.3.2.</i>                              | MicroStation            |

**Tableau 6 : Quelques exemples d'outils métiers pour l'architecture développés sur la base de modeleurs généralistes**

### 2.2.3 Méthode d'analyse de modeleurs choisis

A partir des modeleurs abordés précédemment, nous avons choisi d'analyser plus précisément : Grasshopper, Generative Components, Digital Project et Design Script.

Ce choix est motivé par différents critères, dont les plus déterminants sont :

- la représentativité de ces modeleurs dans l'usage actuel de la modélisation paramétrique. En effet, ces modeleurs paramétriques semblent être les plus utilisés pour la conception architecturale (Kolarevic 2005; Kalay 2004) et, sont mis en œuvre dans les processus analysés dans cette recherche (*cf.* Chapitre 5).
- Les potentialités de ces modeleurs pour la modélisation paramétrique : contrairement aux modeleurs métiers proposant des objets préconçus (*cf.* 2.2.2), ces modeleurs permettent au concepteur de développer ses propres modèles, en définissant de lui-même variables dépendantes et indépendantes, contraintes et relations.

Le modeleur DesignScript, plugin d'AutoCad, a un statut particulier. Il n'est pas encore commercialisé, mais il semble proposer un mécanisme de résolution de contraintes intéressant à expliciter.

Dans l'analyse des mécanismes informatiques de ces quatre modeleurs, nous nous intéressons plus spécifiquement : 1-aux interactions et aux différentes représentations (*explicit*es et *symboliques*) possibles du modèle, 2-au potentiel d'évolution d'un modèle tout au long de sa conception ainsi que, 3-aux compétences sollicitées pour l'usage du modeleur.

- **Les différentes représentations du modèle:** Dans leur article « Multi-Level Interaction in Parametric Design », Aish et Woodbury (2005) pointent une des spécificités majeures de la représentation d'un modèle paramétrique : elle se présente sur différents supports, qui, nous l'avons vu, sont de deux types : *explicite*, représentant la géométrie de façon directement reconnaissable, et *symbolique*, représentant certaines des informations (variables, contraintes, etc.) constituant le modèle. Quels sont ces supports et à quelles informations donnent-ils accès? Comment interagissent ces différentes représentations du modèle (cf. Figure 3 p.25)? sont des points abordés par cette analyse de modeleurs.

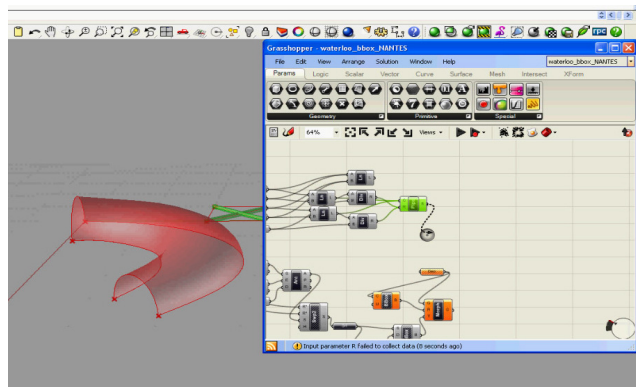
- **Les évolutions possibles du modèle :** Les mécanismes de résolution de contraintes (fonctionnels, variationnels, par système basé sur la propagation, etc.) impliquent des modalités de construction de modèles qui influent sur les possibles évolutions de ceux-ci. Quelles sont les potentialités et les limites de ces évolutions ? est une des questions posées dans l'analyse de ces modeleurs.

- **Les compétences sollicitées :** Représenter un modèle paramétrique sollicite, de la part d'un concepteur, qu'il saisisse le fonctionnement du modeleur : la logique de la propagation de données et de la résolution de contraintes. Il doit également maîtriser des notions de programmation voire des techniques de représentations spécifiques au modeleur (comme un langage de programmation ou des codes spécifiques). La nature des compétences spécifiques sollicitées pour l'usage des modeleurs étudiés sera donc abordée dans cette analyse.

## 2.3 Analyse des modeleurs choisis

### 2.3.1 Grasshopper (GH)

Grasshopper (GH) est un plugin de Rhinocéros développé par l'éditeur McNeel. Un plugin (encore appelé module d'extension ou module externe) est un programme complétant les fonctionnalités d'un logiciel. GH permet de construire une représentation *symbolique* d'un modèle visualisé sur la scène de Rhinocéros (cf. Figure 19).



**Figure 19: Interface de Grasshopper et prévisualisation de la géométrie sur Rhinocéros (de Boissieu & Guéna 2012)**

La représentation *symbolique* construite sur GH fait appel à de la programmation visuelle. La programmation visuelle (ou programmation graphique) est un langage de programmation procédant d'éléments à assembler. Ces éléments sont généralement représentés par des symboles graphiques et « encapsulent » du code, c'est-à-dire des fonctions ou des sous-programmes. En programmation, l'encapsulation est une méthode permettant de protéger un fragment de code. Un code encapsulé dans un composant fonctionne tel-quel et ne peut être modifié. Pour l'utilisateur, ces éléments sont des « boîtes noires » effectuant un traitement dont on ne connaît que le type des données demandées en entrées et produites en sorties. Cette encapsulation simplifie la conception de programme, celle-ci pouvant prendre une forme ludique d'assemblage d'éléments graphiques (cf. Figure 20). La programmation visuelle est beaucoup utilisée comme support pour apprendre à programmer ou pour proposer des environnements de programmation à des non spécialistes de l'informatique. C'est le cas du logiciel « Scratch », développé par le MIT (Massachusetts Institute of Technology) pour permettre à des enfants de programmer de courtes séquences d'animation ou de jeux (cf. Figure 20) ou encore de PureData, un environnement de programmation visuelle développé depuis 1988 pour que des non informaticiens puissent programmer de la musique ou du multimédia en temps réel (cf. Figure 21).

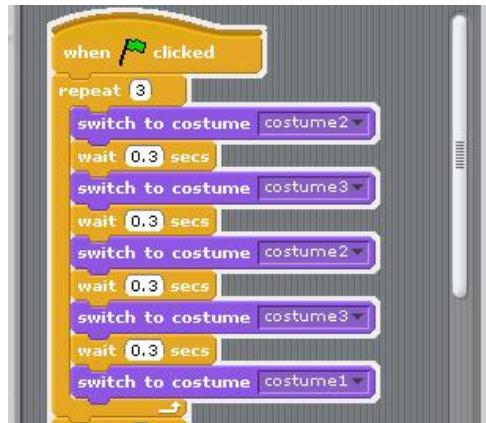


Figure 20 : Exemple de programme développé avec l’environnement de programmation visuelle « Scratch ». Source : [ [scratch.mit.edu/](http://scratch.mit.edu/) ]

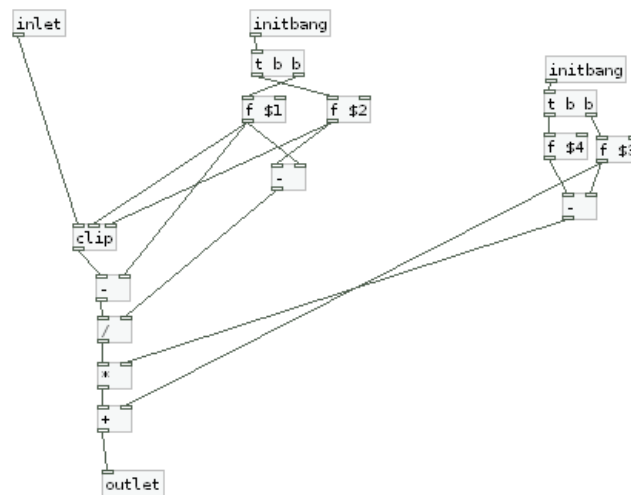


Figure 21 : Exemple de programme développé avec PureData. Source [ [codelab.org](http://codelab.org) ]

Dans le cas de Grasshopper (cf. Figure 19), le programmeur construit un code en reliant les entrées et sorties d’éléments graphiques qui ont des formes de boîtes. Ce sont les « composants ». Il obtient ainsi un graphe dont les nœuds expriment les traitements à effectuer par le programme et les arcs (ou liens) le flux des données. Le programme est exécuté en parcourant le graphe. La modification d’une donnée d’entrée du graphe entraîne une nouvelle exécution du programme. Il s’agit d’un mécanisme de résolution de contraintes par système basé sur la propagation. Les données sont résolues et propagées de proche en proche en suivant la topologie du graphe. Ce graphe est dirigé et acyclique : un composant ne peut être parcouru qu’une seule fois. En cela, même si GH se présente comme un logiciel de programmation visuelle, il s’assimile bien à un mécanisme de modeler paramétrique tel que défini précédemment (cf. 2.1).

- **Spécificités d’usage de Grasshopper sous l’angle des compétences sollicitées**

La programmation visuelle propose un environnement de programmation relativement simple d’accès. Le code étant encapsulé dans les nœuds du graphe, il n’y a pas de risque d’erreur de syntaxe.

Aucune connaissance des structures de contrôles et plus généralement en algorithmique n'est requise. La programmation visuelle est donc un outil de programmation utilisable par des « programmeurs amateurs », c'est-à-dire par des acteurs dont la programmation n'est pas le métier. L'encapsulation des fonctions dans des composants rend l'usage de Grasshopper facile pour les débutants : seules les entrées sont à déterminer. Il n'y a pas de syntaxe spécifique demandée (autre que la composition des composants). Pour un utilisateur plus avancé, certains composants peuvent poser problèmes car peu de documentations existent. Dans ce cas, le forum du site officiel de McNeel pour GH [grasshopper3D.com] peut être une source importante d'informations. Un utilisateur avancé peut également programmer ses propres fragments de codes en C# ou Visuel Basic, ceux-ci prennent la forme de nouveaux composants.

Grasshopper est un système basé sur la propagation. Son usage demande, de la part du concepteur, une maîtrise des flux et des structures des données ainsi que de l'organisation des contraintes. Lors de la conception d'un modèle paramétrique avec Grasshopper, toutes les étapes de construction d'une géométrie sont décrites : le concepteur définit lui-même les contraintes, les variables et leur ordre de résolution.

La structure des données d'entrée d'un composant influe sur l'action de ce dernier (cf. Figure 15 et Figure 16 p.39). La maîtrise de ces structures de données permet de complexifier le modèle et ses traitements : elle permet de faire des traitements itératifs complexes, en ne manipulant toutefois que des graphes dirigés et acycliques (cf. Figure 16 p.39). Ce besoin de maîtrise des structures de données est indispensable à l'usage de Grasshopper. La difficulté de modélisation n'est pas dans la définition des contraintes et des liens entre variables indépendantes et dépendantes, mais elle est dans la structuration des flux de valeurs parcourant le modèle.

#### - **Spécificités d'usage de Grasshopper sous l'angle des interactions possibles avec le modèle paramétrique**

Les modèles développés sur Grasshopper sont visualisables au moyen de l'interface graphique de Rhinocéros. Contrairement à la programmation avec Rhinoscript<sup>9</sup> qui crée des objets « instanciés » définitifs dans la scène de Rhinocéros, Grasshopper crée des prévisualisations d'instances. Un objet créé sur Grasshopper sera visualisé dans la scène de Rhinocéros (cf. Figure 19), mais si le modèle Grasshopper est rendu inactif, la visualisation de la géométrie générée s'arrête. Obtenir un objet définitif à partir d'un modèle paramétrique dans la scène de Rhinocéros est possible grâce à une fonction (appelée « bake ») qui permet de figer l'objet. Cet objet prend alors une existence dans Rhinocéros, coupée du modèle paramétrique qui l'a généré. Ce phénomène de production d'instances « mortes », coupées de leurs historiques, s'observe aussi dans Digital Project.

Aucune modification du modèle paramétrique GH ne peut être faite à partir de sa visualisation *explicite* disponible dans la scène Rhinocéros. Seule des modifications du « programme visuel », constituant la représentation *symbolique* du modèle, peuvent être effectuées. Un surlignage des objets sélectionnés dans le code permet néanmoins de visualiser les différentes représentations d'un objet (cf. objets surlignés en vert dans la Figure 19).

---

<sup>9</sup> Rhinoscript est un langage de programmation spécifique à Rhinocéros. Utilisé originellement *via* le plugin Monkey, le module de programmation est intégré à Rhinocéros depuis sa version 5.

- **Spécificités d'usage de Grasshopper sous l'angle des transformations possibles du modèle**

L'apparente simplicité d'usage de GH (pas de syntaxe à apprendre, visualisation du flux des données) est à nuancer par la rapide complexification du modèle : dès que plusieurs traitements, nœuds et arcs sont mis en œuvre, le modèle peut devenir peu lisible. Ce manque de lisibilité est une difficulté pour faire évoluer le modèle *symbolique*, comme pour son partage ou sa réutilisation (Davis et al. 2011; Harding et al. 2012).

### 2.3.2 Generative Components (GC)

Generative Components (GC) est un modelleur actuellement commercialisé par Bentley Systems. Ce modelleur a initialement été développé par Robert Aish sur la base du projet « Custom Object » (Aish in Kolarevic 2005, p.243-250). Le développement de GC a été marqué par les retours effectués sur son usage lors des workshops organisés par la communauté Smart Geometry (R. Aish & Woodbury 2005). GC est un logiciel qui a évolué et a été défini en interaction<sup>10</sup> avec des professionnels de l'architecture (architectes et ingénieurs) déjà impliqués dans des pratiques professionnelles spécifiques (formes dites « complexes », programmation, structures non standard, etc.) (R. Aish & Woodbury 2005).

GC est un modelleur « basé sur la propagation » (cf. 2.1.3). Cet environnement de programmation est générique et adaptable à toutes industries et conventions (Menges 2006, p.45). Relativement aux caractéristiques des systèmes « basé sur la propagation », GC permet : - de concevoir des chaînes de dépendances et d'associations d'objets permettant de générer des espaces de solutions plutôt qu'une solution spécifique, - de prendre comme nœud de graphe des « données multiples », permettant un traitement des géométries proche de celui permis par les boucles.

GC a la particularité de proposer de multiples représentations *explicites* et *symboliques* d'un même modèle paramétrique. Ces représentations sont autant de support de manipulation du modèle paramétrique (Menges 2006, p.45)<sup>11</sup>. En premier lieu, comme tous modelleurs, GC propose une visualisation de la représentation *explicite* du modèle (cf. Figure 22-A). La géométrie représentée y est modifiable dans une certaine mesure. Par exemple les points sont déplaçables sur la représentation *explicite*, et ces déplacements impliquent en temps réel la modification des coordonnées de ces points dans les représentations *symboliques* disponibles. Ces représentations *symboliques* disponibles sont : -un graphe qui rend compte des nœuds constituant le modèle paramétrique et de leurs relations (cf. Figure 22-B) et -une liste d'action constituant l'historique de construction (cf. Figure 22-C). Cet historique est modifiable : une action peut être transformée, supprimée ou ajoutée *a posteriori*, mais dans la limite de la viabilité du modèle. Ainsi on ne peut utiliser une variable avant de l'avoir créée, ou supprimer un objet qui sera sollicité dans une autre contrainte du graphe.

---

<sup>10</sup> « *The existence of a motivated independant user community and a relatively opened and relatively well-resourced system development process provides opportunities for early and frequent verification of design choices against actual need and for research connecting task to system at a scale largely not possible in research labs using systems built from scratch.* » (R. Aish & Woodbury 2005).

<sup>11</sup> « *representation are an editable, re-executable design history* » (Menges 2006, p.45).



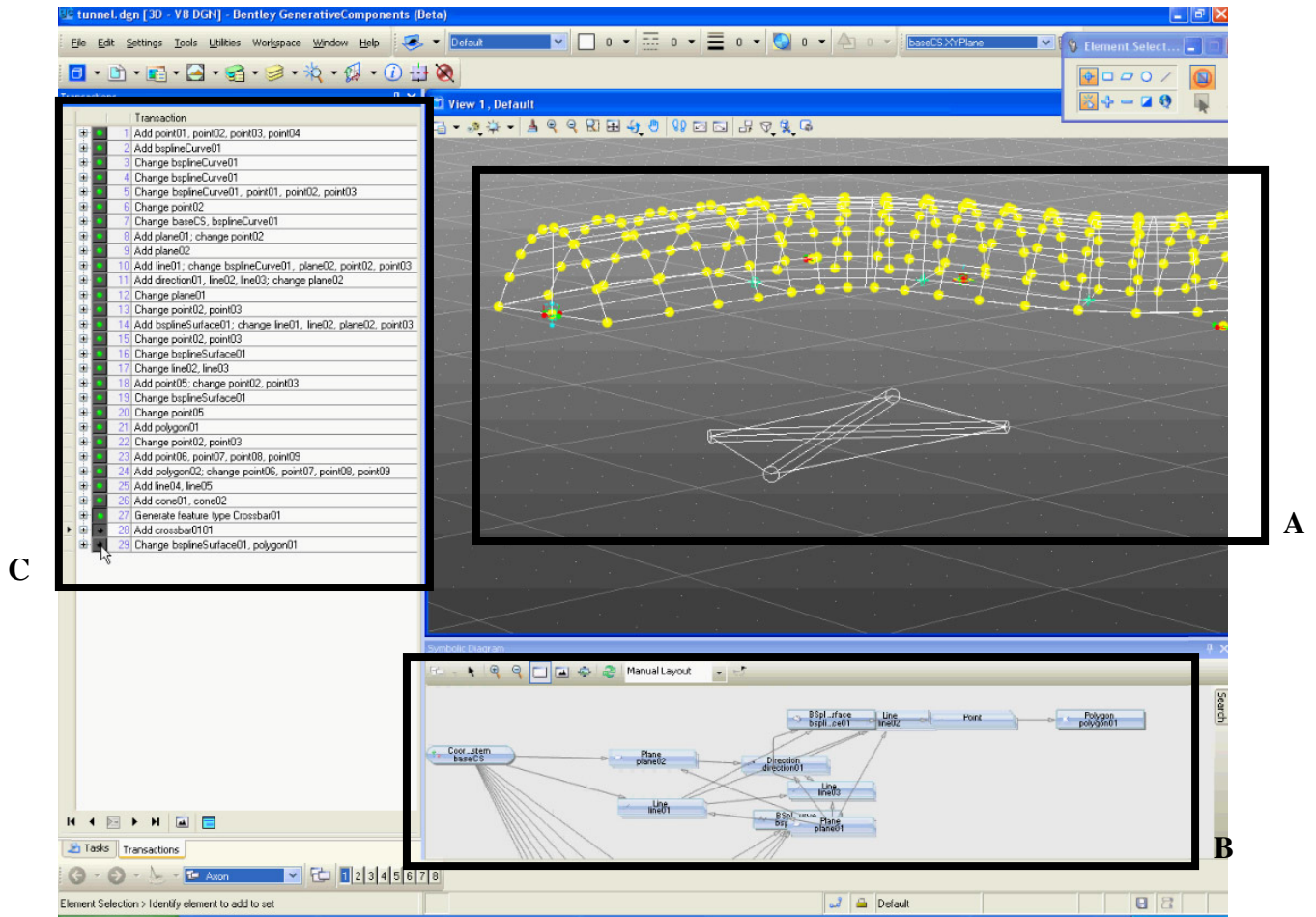


Figure 22 : Exemple de l'interface de GC : une représentation *explicite* (A) et, deux représentations *symboliques* du modèle : le graphe de propagation de données (B) et l'historique de construction (C)

Des fonctionnalités sont rendues accessibles sous forme de menus déroulants. Ces fonctionnalités sont des programmes définis par l'éditeur, comme des méthodes de constructions d'objets géométriques : par exemple «créer un point par des coordonnées cartésiennes», «créer un point en fonction d'un paramètre t et d'une courbe», «créer un point par coordonnées polaires» etc. ou encore des méthodes de manipulation de valeurs, comme les fonctionnalités «fonction à n variables», «moyenne», ou «nombre entier pouvant varier entre des extremums» etc. Au sein de ces fonctionnalités, des fragments de modèles ou des sous programmes développés par l'utilisateur peuvent être intégrés, cela rend ces derniers accessibles pour de futures modèles.

- **Spécificités de l'usage de Generative Components sous l'angle des interactions possibles avec le modèle paramétrique**

Le modèle paramétrique est visualisable et modifiable *via* ses différentes représentations (*cf.* Figure 22 : représentations *explicite* (A) et, *symboliques* (B) et (C)). Cette visualisation multiple permet une grande souplesse dans la manière d'opérer des modifications dans le modèle. Elle multiplie les points de vue sur le modèle et, par conséquent en facilite la lecture. Néanmoins, l'interface reste peu intuitive : les différentes représentations sont cloisonnées même si un surlignage permet de repérer les différentes représentations d'un même objet. Par ailleurs la multiplication des fenêtres demande de manipuler le modelleur sur de grands écrans pour en avoir des visualisations lisibles.

- **Spécificités de l'usage de Generative Components sous l'angle des transformations possibles du modèle**

La manipulation d'un modèle paramétrique sur GC nécessite une bonne compréhension du flux des données. Le modèle est construit par un historique qui n'est pas forcément visible dans toutes les représentations, ainsi il est parfois mal aisé de lire les dépendances liant plusieurs objets. Or le réseau de dépendance établi par le concepteur contraint les évolutions possibles du modèle.

Les évolutions du modèle sont facilitées par la possibilité de programmer, omniprésente dans le modelleur. Ainsi chacune des fonctionnalités préconçues du modelleur est accessible et modifiable par le concepteur. Contrairement à Grasshopper par exemple, où chaque composant est « clos », dans GC les composants nommés « features » peuvent être ouverts et les codes directement manipulés. De la même façon, le code entier du modèle, traduit par le modelleur à partir des opérations effectuées par le concepteur, est accessible et modifiable. C'est finalement la multiplicité des interactions possibles avec le modelleur qui facilite les transformations des modèles (R. Aish & Woodbury 2005).

- **Spécificités de l'usage de Generative Components sous l'angle des compétences sollicitées**

L'usage de GC nécessite certaines compétences en programmation pour comprendre l'organisation des listes ou des collections constituant le flux de données parcourant les graphes. Il suppose également des connaissances du langage spécifique utilisé par GC (un dérivé du C#). Ainsi, si les fonctionnalités sont des fragments de programmes déjà écrits, leurs manipulations nécessitent une courte description de leurs entrées par du code (pour décrire une liste de valeur par exemple). Dans ces cas là, une confusion de syntaxe (une virgule à la place d'un point virgule par exemple) est rédhibitoire dans la construction du modèle.

### **2.3.3 Digital Project (DP)**

Digital Project (DP) est un logiciel métier développé à partir du modelleur CATIA (acronyme de « Conception Assistée Tridimensionnelle Interactive Appliquée » ou « Computer Aided Three-dimensional Interactive Application »). Les origines du développement de CATIA se trouvent dans les années 60' et plus particulièrement dans le développement de CADAM, alors développé par IBM, et qui sera racheté par Dassault Systems (Weisberg 2008). Dès les années 70' le développement du projet « CATI » vise la représentation d'ouvrages de l'industrie aéronautique (Weisberg 2008). Ce projet donne lieu au modelleur CATIA, qui sera plébiscité dans l'aéronautique puis dans de nombreuses autres industries, car il permet de représenter et de contrôler la géométrie d'un ouvrage complexe (en termes de nombre de pièces et de nature de leur géométrie) de la conception à la fabrication. La version 1 de CATIA est proposée en 1981 mais sera utilisée au sein de l'agence Gehry Partners pour la première fois en 1992 par Rick Smith (Smith 2011; Kolarevic 2005)<sup>12</sup>. Cette appropriation et les évolutions des usages de CATIA en architecture seront plus spécifiquement abordées dans le chapitre 3.2.

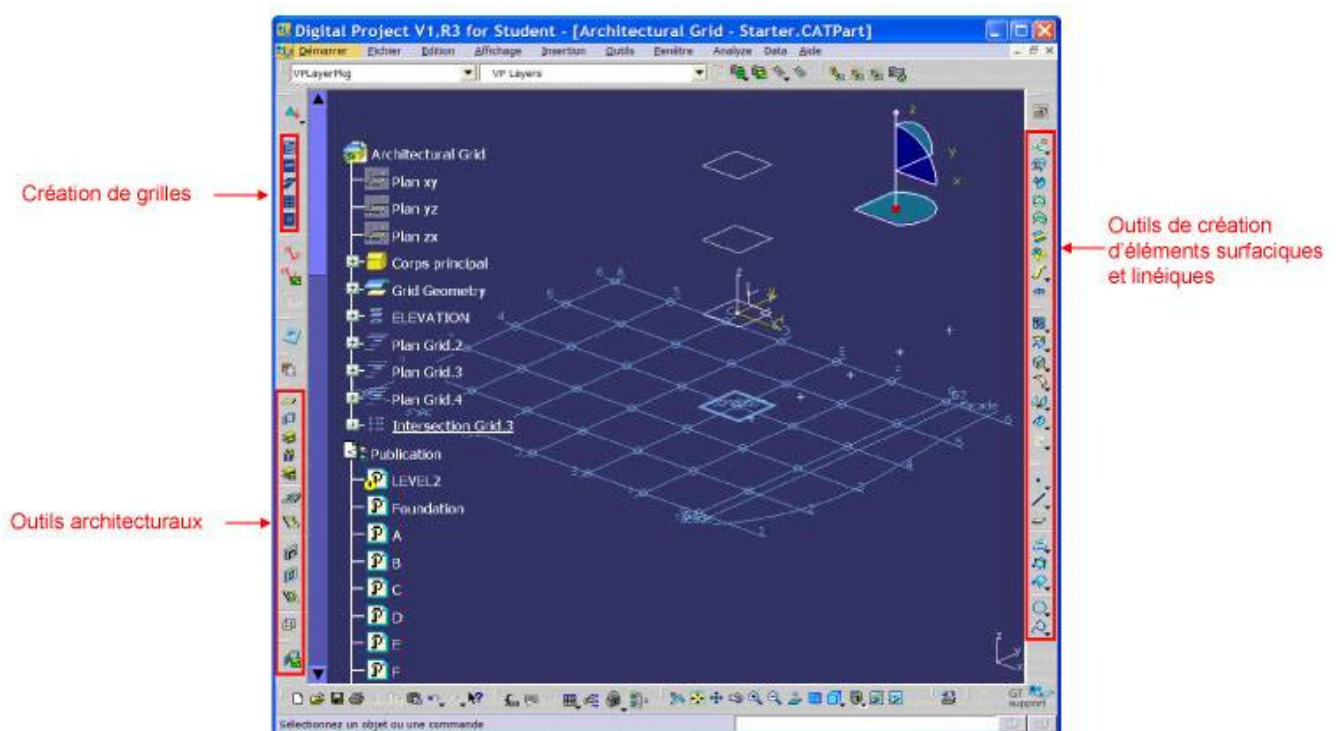
---

<sup>12</sup> « CATIA had been use for 29 years before it was discovered by Gehry's office (and is currently used by very few design offices). » (Kolarevic 2005, p.6).

Gehry Technologies est une entreprise créée en 2003 pour valoriser et développer les savoirs faire numériques de Gehry Partners. C'est elle qui développe Digital Project à partir de CATIA, en vue d'en faire une version métier pour les architectes.

Au sein de CATIA, différents « Ateliers » sont proposés à l'utilisateur. Ces Ateliers sont des configurations de l'environnement de modélisation qui sont adaptés à la réalisation de tâches précises. Il existe des Ateliers métiers : Infrastructure, Mécanique, etc. ; et des Ateliers spécifiques à une phase de travail ou à une fonctionnalité du logiciel, comme par exemple les Ateliers Part Design, Esquisse, Forme, etc. Digital Project propose un « Atelier Architecture » spécifiquement adapté aux besoins des architectes, appréhendés par Gehry Technologies.

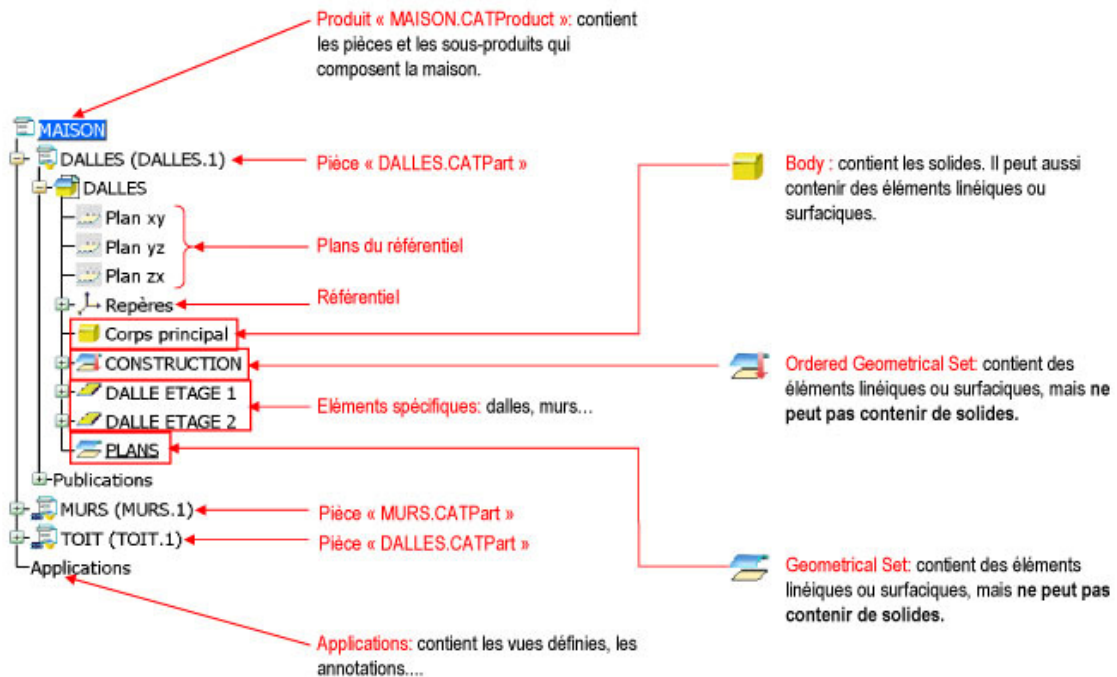
L'atelier « Architecture » (cf. Figure 23) propose : 1-des outils de création de trames, 2-des outils permettant de créer des objets « architecturaux » (poutres, portes, etc.) et 3-des outils de création de lignes et surfaces issues de l'Atelier « Forme ».



**Figure 23: Présentation de l'Atelier Architecture spécifique à DP lors d'une formation de Gehry Technologies, Source : [www.gtwiki.org](http://www.gtwiki.org)**

Comme les autres modelleurs paramétriques, CATIA propose une double représentation du modèle: une représentation *explicite* et une représentation *symbolique*. La représentation symbolique proposée par CATIA prend la forme d'un « Arbre des spécifications » (cf. Figure 24). L'« arbre des spécifications » représente tous les objets utilisés et créés lors de la construction du modèle paramétrique : les géométries créées et leurs méthodes, les variables, les paramètres, les formules, etc.(Mineur 2009). L'arbre des spécifications reprend l'organisation en « Part » et « Product » du modèle (cf. Figure 24). Les « Parts » sont les objets géométriques représentés, ils peuvent être spécifiés plus précisément sous la forme de Set ou de Body en fonction de leur nature (solide, filaire, etc.) (cf. Figure 24). Les « Products » décrivent l'organisation des Parts entre elles. Cette structuration en objets

d'un coté (les Parts) et en structure d'objets de l'autre (les Products) est intéressante pour la réutilisation d'objets (les Parts) dans de nouvelles structures (les Products), ou bien pour travailler avec des fichiers très lourds : dans ces cas là, seules les parts concernées par les opérations d'éditations en cours sont sollicitées, les autres ne sont pas calculées. Elle permet par ailleurs de mettre en place des cinétiques dans le modèle (pour simuler le fonctionnement d'une pièce de moteur ou pour simuler la mise en œuvre d'éléments sur un chantier par exemple).



**Figure 24 : Parts, Products, Set et Body: Extrait de la formation à DP prodiguée par Gehry Technologies, Source : [www.gtwiki.org](http://www.gtwiki.org)**

L'arbre des spécifications peut être, au moment de la caractérisation du modèle, défini comme « ordonné » ou « non ordonné ». S'il est « ordonné », l'ordre des différents nœuds constituant le graphe est contraint par la chronologie de création des objets. Par exemple, si l'on prend l'hypothèse que l'arbre des spécifications de la Figure 24 est ordonné, l'ordre du graphe nous apprend que les objets « Dalles » ne dépendent pas des objets « murs », mais que l'on ne peut probablement pas supprimer ces objets Dalles sans perdre la définition des « murs » et des « toits ».

Si CATIA et Digital Project contiennent un modelleur paramétrique fonctionnel, ils proposent également un modelleur paramétrique équationnel (dit aussi variationnel) au sein d'un module appelé « sketcher » (cf. Figure 25). Celui-ci permet de créer une géométrie en deux dimensions à laquelle on peut ajouter des contraintes *a posteriori*. Cette géométrie plane peut ensuite être utilisée comme un objet (par exemple comme profil, section, emprise en plan etc.) dans un modèle paramétrique en trois dimensions.

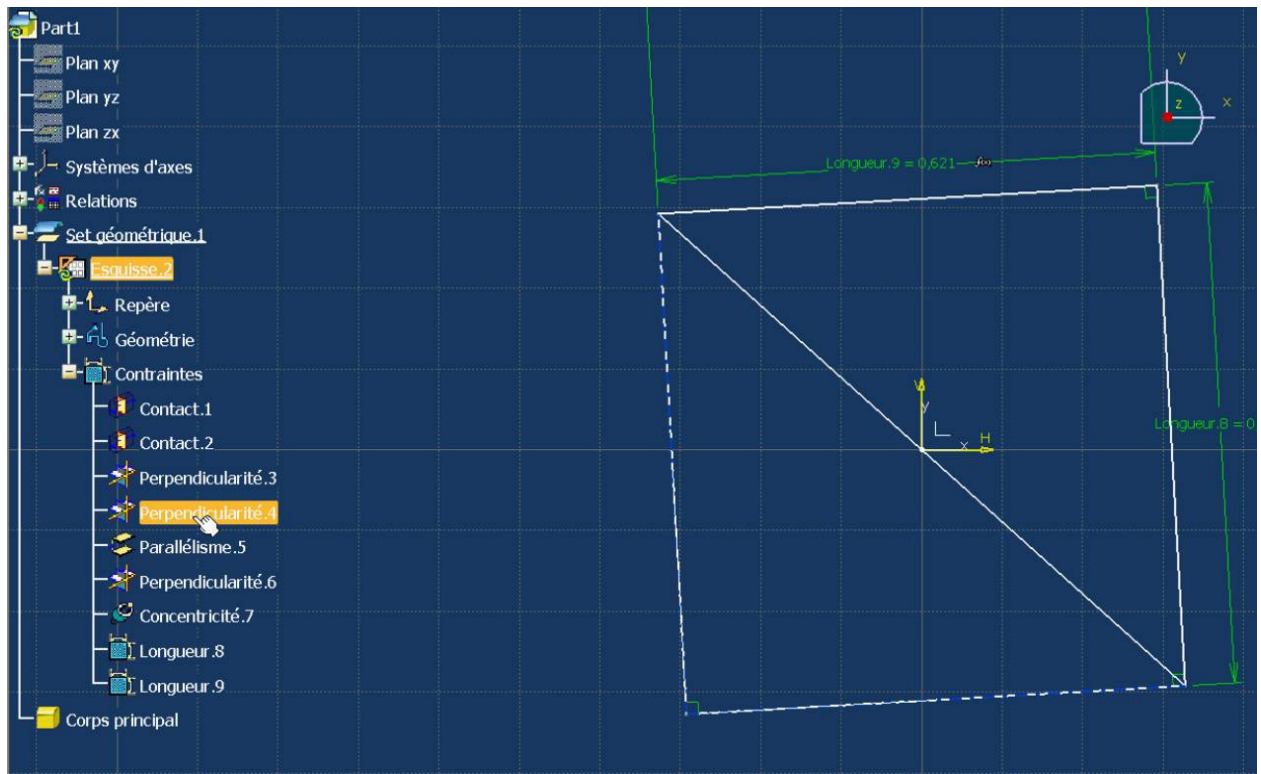


Figure 25 : Interface du "sketcher" ou "mode esquisse" de Digital Project

- **Spécificités d'usage de Digital Project sous l'angle des interactions possibles avec le modèle paramétrique**

Des interactions avec le modèle paramétrique sont possibles : par le sketcher, à partir de l'arbre des spécifications, de la manipulation de la représentation explicite du modèle, ou encore à partir du module « formula » regroupant toutes les formules et toutes les variables dépendantes et indépendantes du modèle dans un tableau.

La résolution des contraintes sur DP n'est activée que sous l'impulsion du concepteur, lorsque celui-ci décide de réactualiser son modèle. Différer le calcul du modèle sur DP permet au concepteur de faire autant de modifications que voulu avant que ne se lancent les nouveaux calculs, ceux-ci pouvant bloquer le modèle ou le déformer de façon inattendue et ainsi le rendre difficilement lisible.

- **Spécificités d'usage de Digital Project sous l'angle des transformations possibles du modèle**

La représentation *symbolique* du modèle sous forme d'arbre de spécification supprime toutes les informations sur les arcs, c'est-à-dire concernant les relations et les associations mises en place entre les différents objets. Seules les informations relatives à l'historique sont conservées dans le cas où l'arbre des spécifications est « ordonné ». Si une méthode de construction d'un objet peut être transformée après sa mise en œuvre, les évolutions plus majeures des modèles restent peu aisées.

#### - **Spécificités d'usage de Digital Project sous l'angle des compétences sollicitées**

L'usage de DP demande des compétences très spécifiques, un savoir technique et une certaine expérience. Outre les compétences en informatique et en géométrie généralement sollicitées pour construire un modèle paramétrique, des compétences spécifiques au logiciel sont ici indispensables à la conception d'un modèle. En effet certains mécanismes sont très spécifiques à ce modèleur, comme l'organisation en Atelier, qui si elle n'est pas comprise entrave l'accès à certains groupes de fonctionnalités. Ou encore l'organisation en parts et en Products, qui doit être comprise pour pouvoir lire l'arbre des spécifications et le modifier au besoin.

#### **2.3.4 DesignScript (DS)**

Design Script est un module d'Autocad dont la commercialisation n'est pas encore effective. La portée d'un tel outil permettant de produire un modèle paramétrique à partir d'un logiciel très utilisé en agence d'architecture mérite qu'on s'y attarde. Le développement de DesignScript est actuellement dirigé par Robert Aish, à qui l'on doit déjà le modèleur Generative Components. Comme Generative Components, DesignScript est développé en laboratoire puis testé auprès de professionnels de l'architecture et de l'ingénierie. Il a déjà été testé au cours de divers Workshop, en particulier lors du Modelling Symposium de Berlin en 2011 et des conférences Advanced Architectural Geometry de Paris en 2012. La volonté de faire de DesignScript un logiciel pédagogique est fortement pointée par Aish (R. Aish 2011; R. Aish 2012). DesignScript serait, selon cet auteur, un logiciel utilisable par des non-experts comme par des experts en programmation. DesignScript est conçu pour être un support d'apprentissage, permettant à des débutants de se familiariser avec les principes de la programmation. Il est également conçu comme une plateforme puissante, permettant à des experts de la programmation et de la modélisation paramétrique de programmer des bibliothèques, de créer des programmes complexes, etc.

DesignScript permet par ailleurs d'importer des géométries ou de créer des liens avec des applications différentes, comme en particulier Robot (pour l'évaluation structurelle) ou Ecotech (pour l'évaluation de l'ensoleillement, *etc.*) (R. Aish 2011, p.4).



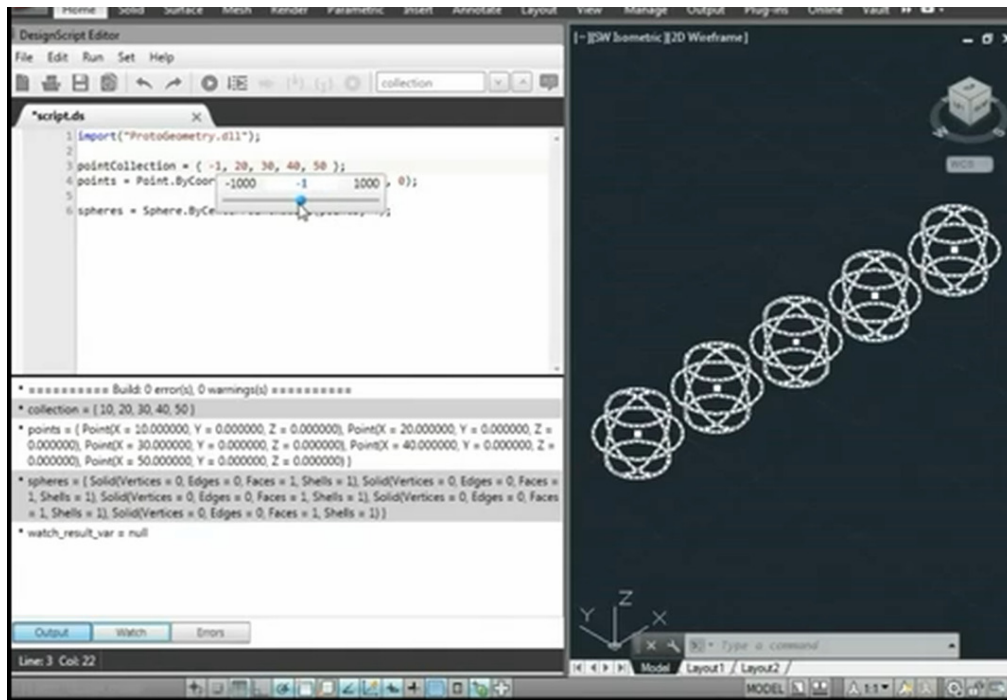


Figure 26 : Interface de DesignScript, source: [labs.autodesk.com]

A ce jour, DesignScript est un module permettant de programmer un modèle paramétrique sur Autocad. En cela, son fonctionnement se rapproche de celui de GH. Cependant, il ne s'appuie pas sur un environnement de programmation graphique mais sur un environnement de Scripting : il propose un éditeur de texte dont les géométries créées sont visualisées sur autocad (cf. Figure 26). Le langage de DesignScript réunit deux paradigmes de programmation : le paradigme associatif et le paradigme impératif. Pour Aish, ce double paradigme permet de réunir la facilité d'usage de l'associatif (grâce à la lisibilité de lecture, la multiplicité des mises à jour de variables en cours de compilation) (R. Aish 2011) et la possibilité d'intégrer la récursivité dans le script de l'impératif (grâce à des procédures du type « if... then... » par exemple) (R. Aish 2012).

Par exemple, dans le manuel de DesignScript proposé sur [labs.autodesk.com] en février 2012, deux modèles sont proposés pour générer une même géométrie de surface (cf. Figure 27 et Figure 28). Les deux modèles entretiennent des points communs : ils ont la forme de scripts, nécessitant de déclarer les variables utilisées et d'appeler les bibliothèques sollicitées. Dans le premier modèle, le script fait appel à un paradigme impératif. Les coordonnées des points constituant la surface sont générées au moyen d'une boucle (cf. cadre pointillé de la Figure 27). Dans le deuxième modèle par contre, le paradigme utilisé est associatif, et les coordonnées des points sont générées liste par liste, puis constituées au moyen d'un produit cartésien (cf. cadre pointillé de la Figure 28).

```

import("ProtoGeometry.dll"); // use the standard DesignScript geometry library
import("Math.dll"); // use the standard DesignScript math library

surfacePoints_2D_array : Point[][]; // define a 2D array of Points
surface : BSplineSurface; // define a surface

[Imperative]
{
  xSize = 10;
  ySize = 15;
  xHeight = 2;
  yHeight = 4;
  numColsX = 8;
  numColsY = 6;

  for(i in 0..numColsX)
  {
    for(j in 0..numColsY)
    {
      surfacePoints_2D_array[i][j] = Point.ByCoordinates( i * (xSize / numColsX), // x coordinates
                                                         j * (ySize / numColsY), // y coordinates
                                                         (Math.Sin(i * (180 / numColsX)) * xHeight)
                                                         + (Math.Sin(j * (180 / numColsY)) * yHeight) ); // z coordinates
    }
  }

  surface = BSplineSurface.ByPoints(surfacePoints_2D_array).SetColor(Color.Cyan); // create a surface
}

```

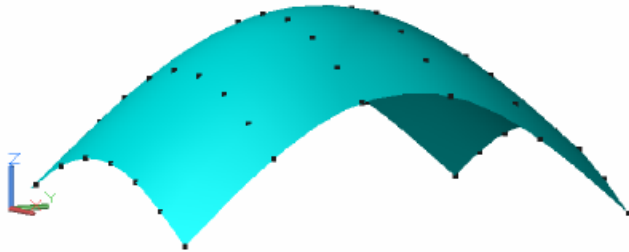
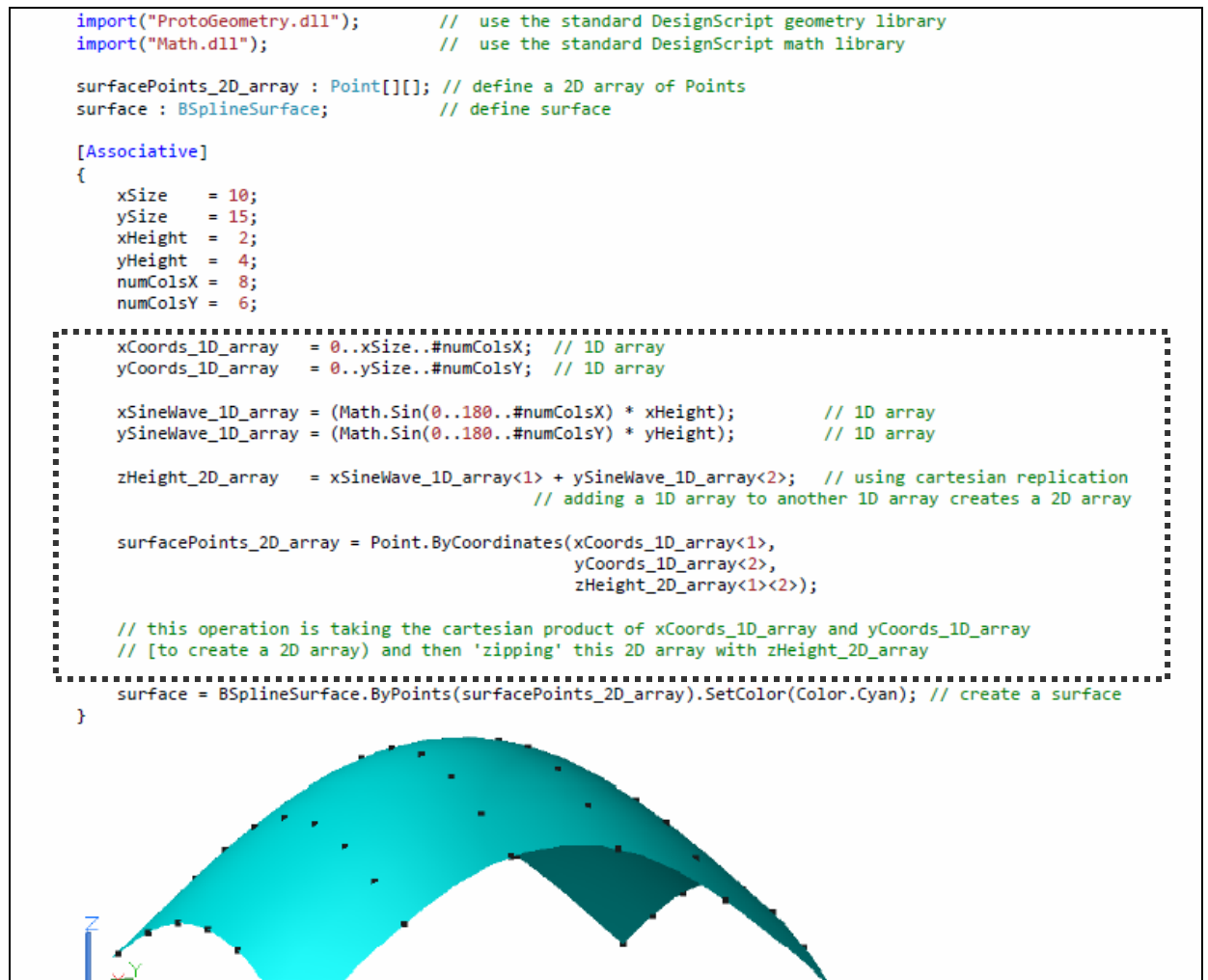


Figure 27 : Modèle sur DesignScript, générant une surface selon un paradigme impératif, source: [labs.autodesk.com]





**Figure 28 : Modèle sur DesignScript, générant une surface selon un paradigme associatif, source: [labs.autodesk.com]**

Cette démonstration fournie par [labs.autodesk.com], montre que les deux paradigmes sont utilisables dans DesignScript, en fonction du modèle visé ou des compétences du concepteur. Le paradigme impératif renvoie aux techniques de scripting usuelles. Le paradigme associatif relève d'un mécanisme « basé sur la propagation ». L'équivalent du modèle associatif reproduit dans la Figure 28 peut être retranscrit sur un autre modèle « basé sur la propagation » comme Grasshopper, comme le montre la Figure 29.

```

[Associative]
{
  xSize = 10;
  ySize = 15;
  xHeight = 2;
  yHeight = 4;
  numColsX = 8;
  numColsY = 6;

  xCoords_1D_array = 0..xSize..#numColsX; // 1D array (1)
  yCoords_1D_array = 0..ySize..#numColsY; // 1D array

  xSineWave_1D_array = (Math.Sin(0..180..#numColsX) * xHeight); // 1D array (2)
  ySineWave_1D_array = (Math.Sin(0..180..#numColsY) * yHeight); // 1D array

  zHeight_2D_array = xSineWave_1D_array<1> + ySineWave_1D_array<2>; // using cartesian replication; // adding a 1D array to another 1D array creates a 2D array (3)

  surfacePoints_2D_array = Point.ByCoordinates(xCoords_1D_array<1>, // (4)
    yCoords_1D_array<2>,
    zHeight_2D_array<1><2>);

  // this operation is taking the cartesian product of xCoords_1D_array and yCoords_1D_array
  // [to create a 2D array) and then 'zipping' this 2D array with zHeight_2D_array

  surface = BSplineSurface.ByPoints(surfacePoints_2D_array).SetColor(Color.Cyan); // create a surface
}

```

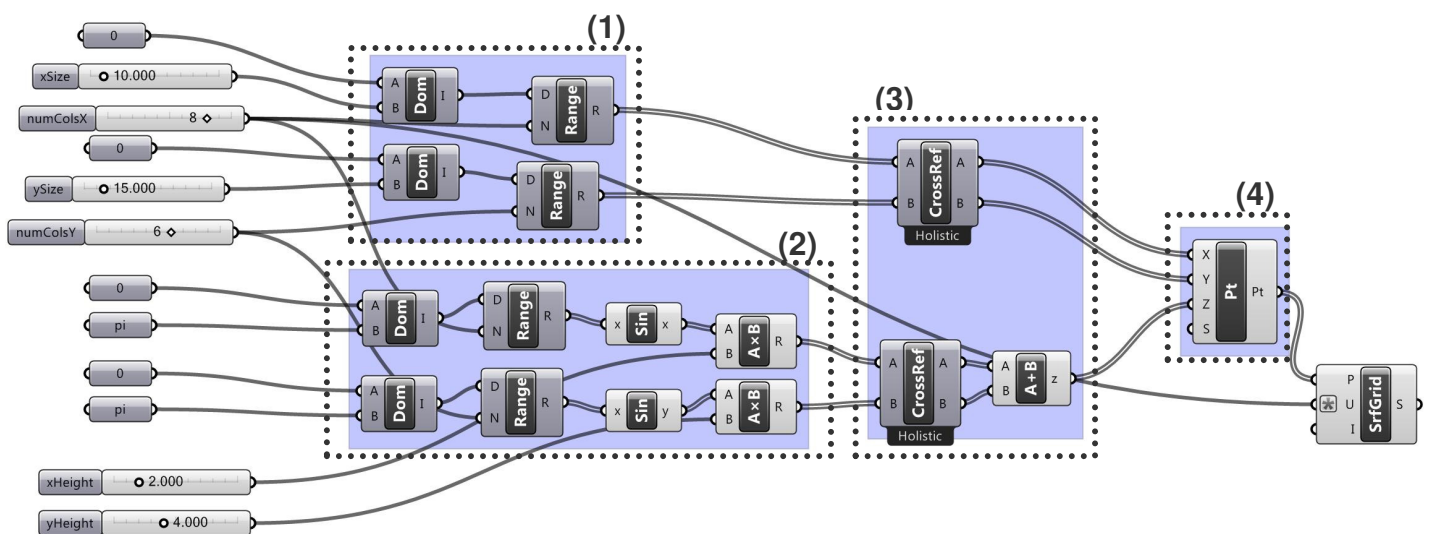


Figure 29 : Equivalence entre un modèle associatif développé sur DesignScript et un modèle développé sur Grasshopper

- Spécificités d'usage de DesignScript sous l'angle des interactions possibles avec le modèle paramétrique

Dans l'état de développement de DesignScript en 2012, la seule interaction possible avec le modèle paramétrique développé est la manipulation du script, *via* l'éditeur. La géométrie générée sur Autocad ne permet pas de modifier le modèle paramétrique lui-même. De plus, si des informations visuelles supplémentaires au script semblent prévues (R. Aish 2012), elle ne sont pas encore disponibles.

- **Spécificités d'usage de DesignScript sous l'angle des transformations possibles du modèle**

L'association des paradigmes associatif et impératif pour créer le nouveau langage de DesignScript vise à faciliter la conception de modèles paramétriques et leurs transformations. Cependant, l'ordre de la définition des variables et des contraintes reste primordial. La transformation d'un modèle existant y est toujours assujettie.

- **Spécificités d'usage de DesignScript sous l'angle des compétences sollicitées**

Il est encore difficile de statuer sur ce modèleur : les différentes présentations qui en sont effectuées (R. Aish 2011; R. Aish 2012) montrent des évolutions radicales de celui-ci (dans le langage créé, l'interface graphique, les mécanismes informatiques en œuvre, etc.). Une première analyse de ce prototype de modèleur montre néanmoins que, si les concepteurs revendiquent la possibilité d'utiliser ce modèleur avec très peu de connaissances en informatiques, celles-ci sont néanmoins nécessaires et, l'apprentissage du nouveau langage de scripting de DesignScript est indispensable (cf. exemples de ce langage dans la Figure 27 et la Figure 28). Ce modèleur vise à rendre possible la programmation de géométries très complexes ainsi que l'interopérabilité de ces géométries avec de nombreux moteurs d'évaluation (Robot, Ecotech et d'autres). Cela rend le modèleur difficile à prendre en main pour un débutant sans prérequis en programmation.

Nous avons vu que nous pouvions définir un modèle paramétrique à partir de : - la présence de plusieurs représentations donnant accès au modèle sous formes *explicites* et *symboliques* ; - la mise en œuvre de mécanismes informatiques permettant de mettre en place des chaînes de dépendances entre les variables et les contraintes constituant un modèle. Cette définition nous a permis d'analyser les modèles utilisés en architecture afin d'en distinguer quatre : Grasshopper, Generative Components, Digital Project et DesignScript. Ces modèles semblent pertinents à analyser dans cette recherche car ils permettent aux concepteurs de définir eux-mêmes les variables dépendantes et indépendantes, les contraintes et les relations constituant les modèles paramétriques conçus, ce qui n'est pas le cas dans tous les modèles.

Si les modèles retenus entretiennent des points communs, ils relèvent en particulier d'un même type de mécanisme « basé sur la propagation », ils ne permettent pas la même lisibilité du modèle et ne sollicitent pas les mêmes compétences en programmation.

Avant d'interroger les activités cognitives en jeu lors de l'usage de ces modèles en conception architecturale, nous abordons dans le chapitre suivant l'usage effectif de ces modèles en agence d'architecture. Dans le cadre de la conception architecturale, qui utilise ces modèles et pour quelles tâches ? est une question abordée par le prochain chapitre.

### **3 Etat des pratiques de la modélisation paramétrique en agence**

Ce chapitre porte sur les pratiques actuelles de la modélisation paramétrique en agence d'architecture. L'analyse de pratiques de quelques agences parisiennes (cf. 3.1) montre que l'usage de la modélisation paramétrique est actuellement en pleine reconfiguration, due, entre autres, au succès rencontré par certains modeleurs ces dernières années, en particulier Grasshopper. Les analyses des pratiques de la modélisation paramétrique chez Gehry Partners (cf. 3.2) et chez Foster and Partners (cf. 3.3) montrent, quant à elles, des pratiques développées depuis longtemps et bien implantées dans les processus des agences.

Dans les différentes analyses menées, la conception de modèles paramétriques concerne des missions variées liées directement ou non à la conception architecturale. Nous le verrons, ces missions concernent, en particulier, la représentation de géométries complexes, l'évaluation des performances structurelles, d'ensoleillement, etc. ou encore de missions liées à la fabrication d'un ouvrage, comme la production de documents d'exécution ou l'optimisation de formes pour les rendre constructibles plus économiquement. Ces missions sont opérées, la plupart du temps, dans le cadre de consultations internes ou externes. La caractérisation du contexte des pratiques de modélisation paramétrique et des missions que cette modélisation paramétrique permet de remplir est le principal objet de ce chapitre.

## **Plan du chapitre :**

|            |   |            |
|------------|---|------------|
| <b>3.1</b> | <b>Modélisation paramétrique et conception architecturale : quelles pratiques ? .....</b>   | <b>65</b>  |
| 3.1.1      | Des pratiques « expérimentales » de la modélisation paramétrique pour la conception architecturale  | 65         |
| 3.1.2      | Des pratiques encore « timides » et cloisonnées   | 67         |
| <b>3.2</b> | <b>Pratiques de consultation externes : le cas de Gehry Technologies.....</b>   | <b>70</b>  |
| 3.2.1      | Intégration du paramétrique au sein de Gehry Partners : Transformation des pratiques entre 1991 et 2003   | 70         |
| 3.2.2      | Le cas de Gehry Technologies Europ et Gehry Partners pour la fondation Louis Vuitton pour la Création   | 84         |
| 3.2.3      | Le cas de Gehry Technologies et AJN pour le projet Qatar National Museum (QNM)  | 89         |
| <b>3.3</b> | <b>Pratiques de consultation interne : le cas de Foster and Partners.....</b>   | <b>92</b>  |
| 3.3.1      | SMG et ARD Team chez Foster and Partners: in-house consultancy  | 92         |
| 3.3.2      | Collaborations et hybridations entre le SMG et les Design Teams   | 97         |
| 3.3.3      | Assistances à la modélisation paramétrique développées chez Foster and Partners   | 97         |
| 3.3.4      | Enjeux de la modélisation paramétrique pour le SMG  | 99         |
| <b>3.4</b> | <b>La profession d'architecte : la modélisation paramétrique comme nouveau métier ou comme extension de l'expertise de l'architecte ? .....</b> | <b>100</b> |
| 3.4.1      | Les pratiques liées à la modélisation paramétrique  | 100        |
| 3.4.2      | Un architecte aujourd'hui   | 103        |

### **3.1 Modélisation paramétrique et conception architecturale : quelles pratiques ?**

Nous nous intéressons ici aux usages de la modélisation paramétrique menées dans quelques agences d'architecture parisiennes. Ces agences ont été retenues pour la représentativité de leurs pratiques. Les pratiques sont présentées ici de façon succincte, en vue d'en esquisser les diversités. Ces pratiques s'avèrent actuellement très mouvantes, c'est pourquoi nous avons finalement focalisé notre analyse sur le cas d'agences étrangères que sont Gehry Partners et Foster and Partners et qui ont des pratiques de la modélisation paramétrique formalisées et stables.

#### **3.1.1 Des pratiques « expérimentales » de la modélisation paramétrique pour la conception architecturale**

Nous nommons ici « expérimentales » des pratiques visant à explorer les potentialités de la modélisation paramétrique pour la conception architecturale. Il semble que la plupart du temps, en France, les pratiques de la modélisation paramétrique soient de l'ordre de l'exploration et de l'expérimentation plutôt que de l'ordre d'une pratique bien établie. Ces pratiques, inscrites dans le champ de l'architecture, donnent actuellement peu de production de bâtiments. Ces pratiques expérimentales produisent en revanche une richesse architecturale sur papier et donnant lieu à des installations.

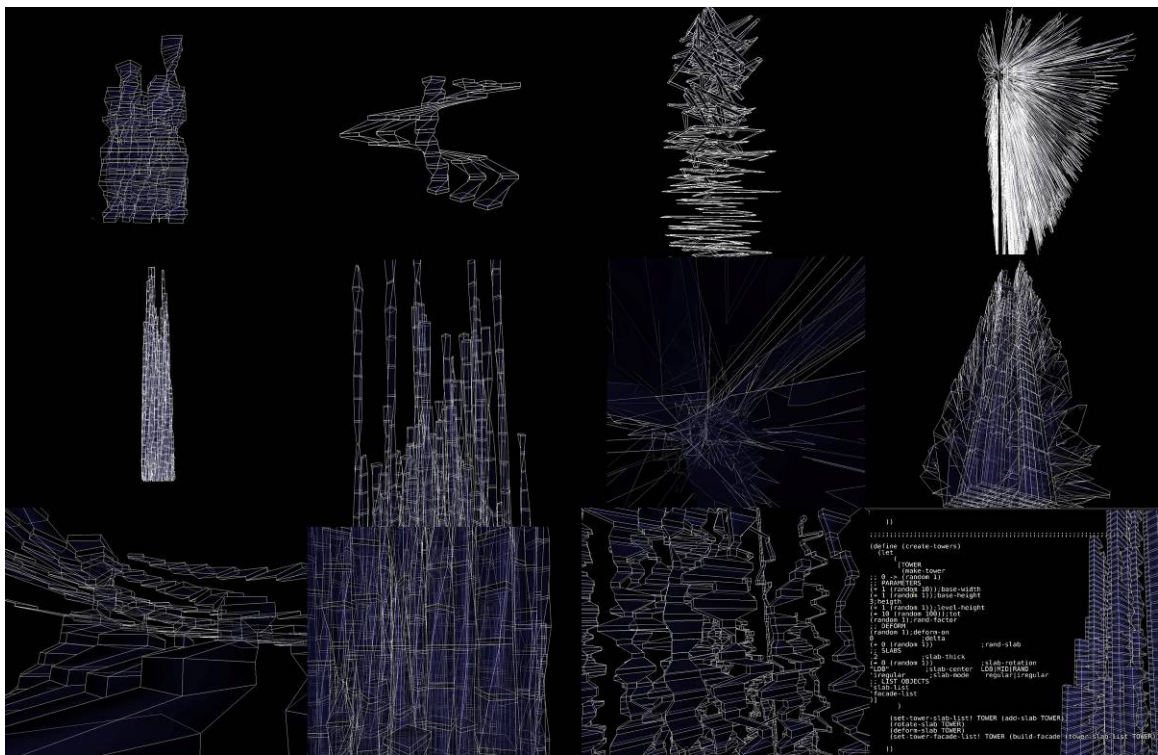
Nous pouvons distinguer trois modalités de ces pratiques expérimentales :

- des pratiques expérimentales développées en parallèle à une activité professionnelle,
- des pratiques expérimentales reconnues,
- des pratiques dites de « *research by design* ».

De plus en plus d'architectes et d'étudiants en architecture s'intéressent à la modélisation paramétrique et aspirent à mettre celle-ci en œuvre dans leur processus de conception architecturale. Néanmoins, ces pratiques restent en marge d'une pratique plus traditionnelle rentable.

L'architecte Svetlin Peev développe ainsi une double pratique architecturale. Il exerce en tant qu'expert de la modélisation paramétrique pour l'aide à la conception et l'exécution de façade et, également, en tant qu'architecte indépendant. Dans le cadre de ces deux pratiques, il utilise la modélisation paramétrique pour concevoir et modéliser l'architecture (Peev 2012). De la même façon, les architectes de « OZCollectif » ont une double pratique de la modélisation paramétrique en conception architecturale: une pratique indépendante au sein de leur collectif et une pratique salariée experte pour des agences parisiennes (Pachiaudi 2011). Pour ces architectes, la réponse à des concours ou le développement de projets personnels sont des opportunités pour expérimenter les outils en dehors des contraintes habituelles.

On trouve une telle posture d'expérimentation *via* l'outil numérique dans l'agence RVBA qui utilise des outils en OpenSource et expérimente l'usage de la programmation pour générer des formes architecturales (Yanatchkov 2010). Ces expérimentations visent la constitution et l'exploration des potentialités de codes informatiques par lesquels générer des géométries. On voit, dans la Figure 30 par exemple, une multiplicité de formes produites à partir d'un même programme informatique. Ces formes procèdent de la manipulation de variables indépendantes. Ce mécanisme est proche de la modélisation paramétrique. Ici, l'architecte a exploré un domaine de solutions défini par son programme, en cherchant ce que Sébastien Bourbonnais nomme le « lâcher prise » (Bourbonnais 2010): c'est-à-dire en cherchant à être surpris par les formes générées. L'architecte affirme ainsi : « *ces formes, c'est comme si ça n'était pas moi qui les avais faites.* » (Yanatchkov 2010). Les expérimentations de l'atelier RVBA ne comprennent pas forcément de visées constructives. Elles semblent plutôt être menées en vue de produire des formes inattendues au travers de programmes génératifs (cf. Annexe p.381). Cette pratique est menée en parallèle d'une pratique plus traditionnelle, permettant aux architectes de vivre.



**Figure 30: Série d'instantanés issus de l'expérimentation pour le concours evolo tower 2009. source : Milovan Yanatchkov**

Ces pratiques visent à explorer les potentialités de la modélisation paramétrique pour la conception architecturale.

Il arrive que des pratiques singulières d'exploration des potentialités de la modélisation paramétrique pour la conception architecturale soient reconnues et financées comme telles.

C'est par exemple le cas, en France, des agences Objectile, EZCT, Theverymany ou ReSie. Ainsi Marc Fornes (de l'agence Theverymany) est intervenu pour des projets de Zaha Hadid et une de ses installations fait partie de la collection permanente du Centre d'Art Moderne Georges Pompidou. De



même, une chaise conçue par Philippe Morel (de l'agence EZCT) fait partie de la collection permanente du FRAC Centre. Développées dans les années 1990-2000, ces pratiques mêlent le scripting et la modélisation paramétrique.

Les travaux de Bernard Cache (agence Objectile) concernant la modélisation paramétrique participent à la construction de ses concepts d'*objectile* et d'*associativité*. Bernard Cache est un des premiers acteurs à s'être intéressé à la modélisation paramétrique pour l'architecture. Il a élaboré avec Gilles Deleuze le concept d'*objectile* (Deleuze 1988, p.26). Un *objectile* s'oppose à un objet fixe, immuable dans le temps et dans l'espace. L'*objectile* est un nouvel objet en transformation continue, qui ne possède pas de forme fixe mais une variation de formes (Deleuze 1988, p.26). Le terme *objectile* a été choisi par Bernard Cache et Patrick Beaucé pour nommer leur agence. Cette agence comprend un atelier de conception architecturale ainsi qu'un atelier de fabrication par usinage du bois. Cache et Beaucé s'intéressent tout particulièrement aux liens entre conception et fabrication. Bernard Cache a développé le concept d'*associativité* qui désigne un continuum entre conception et fabrication (Cache 1997; Migayrou 2003).

Le caractère expérimental de ces pratiques est premier. Il est intéressant de noter la grande proximité entre ces praticiens, acteurs de la modélisation paramétrique et les éditeurs de logiciels ou les enseignants et les chercheurs inscrits à l'université. En effet, de nombreuses explorations et avancées de l'usage de la modélisation paramétrique pour la conception architecturale s'observent au sein de communautés d'architectes participant à une « recherche par le projet » (dite « research by design »). C'est par exemple le cas de Corneel Cannaearts en Belgique. Cette pratique de conception consiste à explorer des potentialités de l'outil numérique au sein de pratique d'enseignement et d'une recherche par le projet (Cannaert 2011). Cette « recherche par le projet » consiste principalement en un retour réflexif et prospectif sur une pratique personnelle de l'architecture. Les discours produits portent sur des pratiques de conception personnelles ne revendiquant pas de *rupture épistémologique*. Si nous ne remettons pas en cause l'intérêt des discours produits, il est à noter que la « recherche par le projet » est opposé à la recherche porté par le présent travail comme par le champ de l'architecturologie en général (Boudon 1971; Lecourtois 2006b).

### 3.1.2 Des pratiques encore « timides » et cloisonnées

De nombreuses tentatives pour faire entrer la modélisation paramétrique dans une pratique de projet existe. Néanmoins, en France, ces pratiques sont encore timides. La plupart du temps, elles sont mises en œuvre par des individus ou des équipes « expertes » de la modélisation paramétrique qui interviennent en consultance au sein du projet.

Un « expert » est entendu ici comme un professionnel reconnu, disposant de compétences spécifiques et reconnues (Lecourtois et al. in de Coninck & Deroubaix 2012). Nous nous intéressons ici à des pratiques savantes, exercées dans un cadre professionnel.

Des pratiques expertes de la modélisation paramétrique s'observent:

- sur le mode de la consultation externe, par exemple au sein de Renzo Piano Building Workshop,
- sur le mode de la consultation interne, par exemple au sein des agences Ateliers Jean Nouvel ou Hugh Dutton Associates.

La consultation externe est courante en architecture et consiste à faire appel à un prestataire pour une expertise dont l'agence commanditaire a un besoin ponctuel. C'est ce qu'a fait l'agence Renzo Piano Workshop Building (RPWB) pour remplir une mission de modélisation paramétrique dans le cadre du projet de la Fondation Jérôme Celu Pathé (Pachiaudi 2011). Le consultant sollicité, Arnold Walz de l'agence DesignToProduction, avait une mission à la fois de représentation, d'aide à la conception et d'aide à la fabrication (Pachiaudi 2011). Il s'agissait pour Walz de représenter l'enveloppe extérieure du bâtiment à partir de la géométrie conçue par RPBW. A partir de cette modélisation, Walz devait concevoir un calepinage de panneaux de verre à partir des éléments définis par RPWB (rythme, orientation de la trame et détails constructifs). Cette utilisation de la modélisation paramétrique par consultation externe est en cohérence avec la culture de l'agence qui externalise beaucoup de missions (Pachiaudi 2011).

Ateliers Jean Nouvel (AJN) est une des agences parisiennes comptant le plus grand nombre de collaborateurs et de projets à grande échelle. Si cette agence fait parfois appel à des consultants extérieurs pour des missions de modélisation paramétrique, elle a également mis en place sa propre pratique du paramétrique en interne. Dans un mémoire, développé au sein du Séminaire Conception Digitale de l'ENSA Paris la Villette dirigé par François Guéna et Caroline Lecourtois, Toshihiro Kubota, un des Partenaires d'AJN, présente l'implication de l'informatique sur l'architecture de l'agence. Il relève qu'il faut attendre 2010 pour voir se constituer une «Cellule de Conception 3D», visant à produire des représentations 3D, des scripts et des modèles paramétriques (Kubota 2012, p.33). AJN utilise le modelleur paramétrique Digital Project depuis 2008 (concours du Qatar National Museum, cf. 2.3.3). Cette pratique est menée avec l'aide de l'agence Gehry Technologies Europ, intervenant comme consultant externe (ce cas est analysé plus précisément dans le paragraphe 3.2.3). Depuis cette collaboration, deux licences de Digital Project sont conservées chez AJN. En 2012, le logiciel n'est plus utilisé (Kubota 2012, p.41) mais, le logiciel Grasshopper fait son entrée sur des tâches circonscrites et spécialisées, liées à l'évaluation ou à la représentation (Kubota 2012, p.43). Au sein d'AJN, la modélisation paramétrique est une pratique menée par une équipe distincte des acteurs de la conception.

Si AJN ne revendique pas un intérêt pour la modélisation informatique (Nouvel 2010), il n'en est pas de même pour Hugh Dutton, responsable de l'agence Hugh Dutton Associates (HDA). HDA est une agence d'architecture et d'ingénierie qui s'intéresse beaucoup aux outils numériques, comme le montre le développement de son blog : [complexitys.com]. Les membres d'HDA seraient « *ni architectes, ni ingénieurs, mais un peu des deux* » (Dutton & Cingolani 2011). Cette agence, de moins de dix collaborateurs, regroupe des profils d'ingénieurs, d'ingénieurs/architectes et d'architectes.

L'agence mène deux sortes de mission :

- des consultations pour des projets « déjà conçus » : c'est le cas de leur mission pour le Département des arts de l'islam conçu par l'architecte Rudy Ricciotti en 2012,
- des projets personnels, qui prennent souvent la forme d'ouvrages d'art : c'est le cas du projet « Footbridge » de Turin en 2006 ou encore d'une verrière pour un centre commercial à Dazongli en 2011.

Dans ce dernier cas de mission, conception et modélisation sont menées conjointement au sein de l'agence. Pourtant, bien que ces deux missions soient menées au sein d'une même structure, il ne semble pas qu'elles le soient par les mêmes acteurs. En effet, si la taille réduite de l'agence permet une bonne communication et des échanges rapides entre les compétences de chacun, chaque membre garde malgré tout sa « spécialité » (la programmation, l'évaluation des structures, le développement d'un projet, *etc.*). Les acteurs de la modélisation paramétrique ne sont pas les acteurs de la conception architecturale.

Ces quelques pratiques de la modélisation paramétrique au sein d'agences d'architecture établies, témoignent d'une évolution de l'inscription de la modélisation paramétrique dans ces agences avec l'arrivée de nouveaux collaborateurs ou d'étudiants stagiaires.

Pour étudier plus précisément les usages et les enjeux de la modélisation paramétrique en agence d'architecture, nous analysons dans la suite de cette thèse deux pratiques d'agences où la modélisation paramétrique est utilisée depuis plusieurs années: Gehry Partners et Foster and Partners.

## **3.2 Pratiques de consultation externes : le cas de Gehry Technologies**

Développées depuis les années 1990, les pratiques de modélisation paramétrique au sein de l'agence Gehry Partners sont bien documentées et permettent de cerner leurs enjeux professionnels. Dans un premier temps, nous analysons l'intégration de la modélisation paramétrique dans le processus de l'agence (3.2.1), entre l'arrivée de Rick Smith en 1991 jusqu'à 2003 et la création de « Gehry Technologies ». Cette analyse permet d'identifier les résistances et les transformations du processus de conception face à l'usage de la modélisation paramétrique. Elle permet également de décrire les missions remplies par la modélisation paramétrique (représentation, évaluation, rationalisation, aide à la fabrication), en lien avec la conception architecturale. Dans un second temps, deux missions de modélisation paramétrique effectuées par Gehry Technologies sont analysées : l'une pour le Pavillon de la Fondation Louis Vuitton pour la Création de Gehry Partners de 2006 à 2013 (3.2.2) et, l'autre pour le Qatari National Museum des Ateliers Jean Nouvel en 2008 (3.2.3).

### **3.2.1 Intégration du paramétrique au sein de Gehry Partners : Transformation des pratiques entre 1991 et 2003**

La présente analyse se base sur les récits de Rick Smith, premier utilisateur de la modélisation paramétrique au sein de l'agence et, de Dennis Shelden, co-fondateur de Gehry Technologies en 2003. Les documents dont nous disposons sont : l'ouvrage, non publié, *Changing the shape of architecture digitally* de Rick Smith (Smith 2005) qui décrit chronologiquement son expérience au sein de l'agence de 1991 à 2003 ; un entretien de ce dernier, mené pour cette recherche (Smith 2011), ainsi que la thèse de doctorat de Dennis Shelden publiée au MIT en 2002. Ces données sont complétées par : les travaux de Bruce Lindsey publiés dans *Digital Gehry, Material Resistance Digital Construction* (Lindsey 2001), les articles « Evolution of the digital design process » de Jim Glymph (Kolarevic 2005), « Adapting Catia » de Steele publié dans l'ouvrage *Architecture and Computers: Action and Reaction in the Digital Design Revolution* publié en 2002 ainsi que, par diverses interviews de Frank Owen Gehry publiées dans la presse (Gehry 2002; Gehry 2007; Vignal 2010).

Cette analyse présente, dans un premier temps, l'intégration de la modélisation paramétrique dans l'agence et, dans un second temps, les enjeux de ces pratiques et leurs impacts sur la conception architecturale chez Gehry Partners.

#### **3.2.1.1 Modélisation paramétrique: missions remplies et inscription dans les processus de l'agence**

##### **3.2.1.1.1 Premières intégrations : les projets Fish (1992) et Walt Disney Concert Hall (1987-2003)**

En 1962, Frank Owen Gehry fonde son agence Frank Owen Gehry and Associates (FOGA) à Los Angeles. En 2000, cette agence change de nom et devient Gehry Partners LLP. Pour faciliter la lecture de ce texte, nous ferons référence à l'agence sous son appellation actuelle (Gehry Partners) tout au long de ce document.

Au début des années 1990, Gehry Partners produit des architectures marquées par des formes très caractéristiques, incluant de nombreuses surfaces courbes, comme au Centre Américain de Paris (l'actuelle Cinémathèque Française). A ce moment là, tous les dessins étaient produits à la main (Smith 2005, p.41) ce qui posait problème quant à la production des documents d'exécution et de construction (Shelden 2002, p.26). Pour représenter les projets, les architectes extraient des représentations 2D à partir des maquettes physiques suivant deux méthodes : en générant l'ombre des maquettes sur des supports blancs au moyen de fortes lumières, ou en plaçant la maquette dans une boîte quadrillée pour prendre les mesures des altitudes des points remarquables au moyen d'un fil à plomb (Smith 2005, p.41).

Dans un premier temps, le projet «Fish» et le projet du Walt Disney Concert Hall ont été développés grâce à ces techniques. Mais, face à l'ampleur des difficultés de représentation et de construction rencontrées, Frank Gehry décide de recourir aux technologies numériques. En 1991, il n'y a aucun ordinateur dans l'agence (Glymph in Kolarevic 2005, p.105). N'étant lui-même pas très averti de ces technologies, Frank Gehry embauche un nouveau collaborateur dans ce but (Smith 2005, p.39). Ce collaborateur, Jim Glymph, est architecte. S'il ne manipule pas lui-même des modeleurs numériques, il est très intéressé par ces techniques.

Après plusieurs investigations auprès des principaux acteurs de la modélisation, Jim Glymph contacte un consultant d'IBM spécialisé dans la modélisation paramétrique sur CADDAM et CATIA : Rick Smith. Smith n'est pas un architecte mais un ingénieur en mécanique, qui a l'expérience de diverses industries, aérospatiales, médicales ou automobiles (Smith 2005, p.20). Il a été contacté par Gehry Partners pour participer à la conception du projet Fish (Smith 2005, p.41).

Le projet Fish, construit en 1992, est une installation du Village Olympique de Barcelone. Il s'inscrit dans un complexe résidentiel et commercial dessiné par les agences Skidmore Owings and Merrill (SOM) et Gehry Partners. La sculpture de 54 mètres de long et 35 mètres de haut fait référence à la forme d'un poisson. Sa construction en acier se compose d'une ossature et d'une peau tissée de bandes métalliques (cf. Figure 31). Quand Rick Smith est contacté, la forme conçue par Gehry au moyen d'esquisses est représentée *via* une maquette physique et certains documents descriptifs produits en 2D. En 1991, l'équipe de conception est alors dans une phase où elle est confrontée à la difficulté de produire les documents nécessaires à la construction de l'objet (Lindsey 2001, p.34).

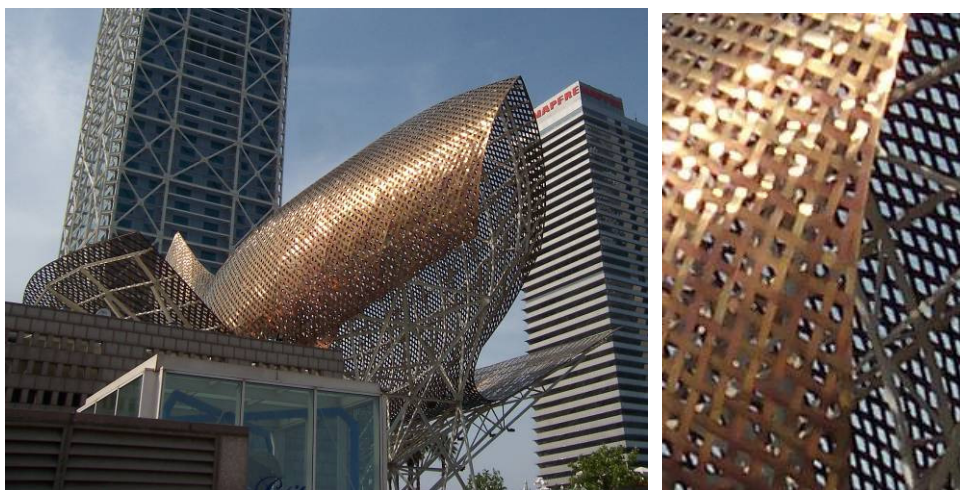


Figure 31: Photographies du projet Fish réalisé, source: [worldalldetails.com]

Smith est sollicité par Glymph pour modéliser le projet sur CATIA V3. A partir d'une section du projet et d'une maquette physique au 1/8e du poisson, Smith modélise une surface représentant l'enveloppe du projet (Smith 2005, p.43). Le modèle paramétrique est validé et affiné de façon itérative avec les architectes (Smith 2005, p.43). Après validation par Frank Gehry, le modèle est transmis aux ingénieurs chargés d'évaluer et de dimensionner la structure. Ce modèle est par la suite utilisé par Smith pour modéliser le calepinage de la peau métallique (Smith 2005, p.44). Le calepinage devait à la fois répondre aux contraintes techniques et constructives du choix des matériaux et de leurs mises en œuvre : des bandes d'acier croisées (cf. Figure 31), tout en répondant aux intentions esthétiques voulues par Gehry. Plusieurs aspects de la conception du modèle paramétrique peuvent être relevés :

- la conception du modèle paramétrique du calepinage prend en compte la façon dont l'objet sera vu pour correspondre aux intentions esthétiques du projet, ces intentions esthétiques servent de « modèle » à la conception de la géométrie paramétrique (du point de vue architecturologique, cela pourrait être interrogé en termes d'*échelle de modèle* ou d'*échelle optique*<sup>13</sup>, cf. Chapitre 1.3)
- la conception du modèle paramétrique prend en compte des considérations techniques pour modéliser des surfaces facilement constructibles à partir de bandes d'acier entremêlées (du point de vue architecturologique, cela pourrait être interrogé en termes d'*échelle technique*<sup>14</sup>, cf. Chapitre 1.3),
- la conception du modèle paramétrique du calepinage se rapporte à la spécification d'une emprise de l'objet conçu : ici celle-ci est la surface de référence constituant l'enveloppe du projet, conçue par Frank Gehry (du point de vue architecturologique, cela pourrait être interrogé en termes d'*échelle parcellaire*<sup>15</sup>, cf. Chapitre 1.3).

L'organisation du modèle paramétrique proposée par Rick Smith, distingue une peau, un espace de connexion et un espace pour la structure (Smith 2005, p.44). Cette organisation facilite la modélisation du bâtiment en rendant possible la modification de la surface de la peau et la visualisation des espaces réservés aux connexions et à la structure. La conception de cette organisation du modèle paramétrique est ici orientée par l'usage du modèle (ce qui, en termes architecturologiques correspond à une *échelle fonctionnelle*<sup>16</sup>). Pour Lindsey, cette organisation du modèle paramétrique est courante dans l'industrie automobile où les modèles sont développés comme des « enveloppes spatiales » de structure (Lindsey 2001, p.37). Cette organisation relèverait donc également d'une *échelle de modèle*<sup>17</sup> : c'est-à-dire que le modèle paramétrique a été conçu en reprenant une référence d'organisation antérieure. Cette organisation du modèle paramétrique par peau, connexion et structure s'observe également dans les

---

<sup>13</sup> L'*échelle optique* correspond à l'activité de « tenir compte d'un point de vue, au sens propre, pour donner à une partie ou à un tout de l'espace architectural une mesure qui fasse intervenir les modalités suivant lesquelles il est vu » (Boudon et al. 2000, p.167).

<sup>14</sup> L'*échelle technique* consiste à « Utiliser des considérations d'ordre technique pour induire une modalité d'attribution de mesures à une partie ou un tout de l'espace architectural » (Boudon et al. 2000, p.175).

<sup>15</sup> L'*échelle parcellaire* correspond à « Mettre en œuvre des mesures en utilisant les possibles permis par la taille, la forme et les limites du terrain donné à l'architecte. » (Boudon et al. 2000, p.172).

<sup>16</sup> L'*échelle fonctionnelle* relève de l'activité de donner de la mesure à un objet « en correspondance à quelque élément extérieur qui en règle la destination, l'utilisation, ou l'adapte à un usage » (Boudon et al. 2000, p.168).

<sup>17</sup> L'*échelle de modèle* implique de « Reprendre un modèle antérieur, tout en effectuant éventuellement des modifications de divers degrés et de diverses natures » (Boudon et al. 2000, p.177).

projets du musée Guggenheim de Bilbao et l'Experience Music Project de Seattle (Lindsey 2001, p.35). Ces « organisations de modèles » seront interrogées dans le Chapitre 6 grâce au concept architecturologique de *découpage*<sup>18</sup>.

Une fois la conception du calepinage et sa modélisation sur CATIA validée par Frank Gehry, la construction du projet Fish a pu commencer. La structure ainsi que le calepinage ont été construits par l'entreprise italienne Permasteelisa. La production des pièces a été faite à partir des représentations fournies par CATIA. Pour ce projet, Glymph a convaincu Permasteelisa de collaborer en interne avec Rick Smith pour la production des données nécessaires à la fabrication (Smith 2005, p.45). La mise en œuvre des pièces sur le chantier a également été effectuée au moyen du modèle paramétrique, grâce à la production de feuilles de calcul décrivant les relations entre chaque pièce. La réalisation du projet Fish est considérée comme un grand succès par Glymph<sup>19</sup> et Smith car les contraintes de délais, de coûts et de géométrie ont été respectées (Shelden 2002, p.29).

Suite au succès de Glymph et Smith pour le projet Fish, Frank Gehry commande une modélisation paramétrique pour le Walt Disney Concert Hall (WDCH). Le WDCH est un projet privé réalisé à Los Angeles sur les fonds de Lilian Disney en hommage à son mari. Le concours organisé pour attribuer le projet a été remporté en 1987 par Gehry Partners. Nous le verrons plus loin, le projet subira de nombreux contretemps. En 1992, la conception du projet est bien avancée, cependant la constructibilité du parement en pierres de la façade est mise en doute par Frank Gehry. Fort de sa réussite pour le projet Fish, Glymph lui propose de tester ici l'utilisation de CATIA (Smith 2005, p.49).

Lors de la modélisation des façades du WDCH, le premier problème soulevé est la numérisation des maquettes physiques conçues par Gehry (Smith 2005, p.49). Pour numériser les géométries construites en papier et carton, Reddy et Smith explorent les techniques de numérisation disponibles. Les numérisations par le son ou la lumière s'avèrent trop coûteuses et nécessitent des conditions de mise en œuvre difficilement conciliables avec l'environnement des ateliers maquettes (Smith 2005, p.49). La numérisation par scanner 3D génère des données considérées par Smith comme trop nombreuses et trop incomplètes pour être exploitables : en effet le scanner ne permet de relever que ce qui est visible et directement atteignable par sa station émettrice (Smith 2005, p.50). C'est finalement une technique de numérisation par un bras mécanique articulé (appelé FARO arm, cf. Figure 32) qui est retenue. Cette technique est utilisée par un médecin pour relever des points sur l'œil, en vue d'en appréhender la forme avant une opération (Smith 2005, p.50). Ce système a permis de numériser la maquette physique du WDCH (cf. Figure 32).

---

<sup>18</sup> Voir la définition du terme architecturologique « *découpage* » dans le paragraphe 5.1.2 p.145.

<sup>19</sup> En 2004, Glymph affirme ainsi: « *Our initial experiments in this were astoundingly successful. It shocked us all. We did our first paperless process in 1991 on a fish sculpture in Barcelona. It created a wonderful collaborative environment between us and the fabricators and builders who executed the work. The project was finished perfectly to what Gehry was looking for, ahead of schedule, on budget and everyone enjoyed the process, which was one of the most important things.* » (Glymph in Smith 2005, p.46).



**Figure 32: Numérisation de points d'une maquette physique au moyen d'un bras articulé, image publiée dans (Smith 2005, p.50)**

A partir de la modélisation des surfaces du projet, Smith a calepiné le parement en pierres de la façade extérieure. Smith avait déjà effectué de telles missions dans le champ de l'aéronautique (Smith 2005, p.51)<sup>20</sup>. Pour mettre en place le calepinage, il évalue les formes des surfaces non réglées et non régulières. Ces géométries impliquent un découpage onéreux de pierres toutes différentes (Smith 2005, p.51). C'est la raison pour laquelle Smith a proposé une nouvelle géométrie, recomposée à partir de fragments de géométries régulières (comme des portions de tores, de cylindres, *etc.*)<sup>21</sup>. L'objectif de Smith était le suivant : « *my goal was to demonstrate the ability of finding surfaces that closely fit the original paper model, yet would produce rationalized surfaces and repetitive families of similar stone panels.* » (Smith 2005, p.51). Dans le cadre de la conception de ce modèle paramétrique, on observe la mise en œuvre d'exigences visant à ce que les nouvelles surfaces produites correspondent aux surfaces de référence (ce qui pourrait relever d'une *échelle parcellaire*). L'usage d'une surface de référence est opéré dans le modèle paramétrique, en vue de faire correspondre les objets modélisés avec les surfaces conçues sur maquette physique (ce qui pourrait relever d'une *échelle optique* ou de *modèle*).

Des aspects de la conception du modèle paramétrique croisent la conception architecturale elle-même. En effet, les nouvelles surfaces sont définies géométriquement à partir de portion de solides euclidiens (*échelle géométrique*) de façon à pouvoir être construites facilement (ce qui pourrait correspondre à une *échelle technique* ou *fonctionnelle*) et pour un budget le plus réduit possible (*échelle économique*).

Cette modélisation paramétrique produit ici de nouvelles surfaces, visant à optimiser, du point de vue de leur constructibilité, les surfaces précédentes conçues par les architectes. Cette reconstruction ne faisait pas partie des habitudes de l'agence qui adaptait le financement aux nécessités de la construction. Dans un premier temps, cette transformation des formes fut mal perçue par les membres de l'équipe de

<sup>20</sup> « *This was reminiscent to me of the method of paneling the space shuttle's underbelly, applying a standardized panel scheme on an undulating surface.* » (Smith 2005, p.51).

<sup>21</sup> Cette expérience, Smith la décrit ainsi: « *Looking at the geometry, I could see a way to break the surfaces down into segments of rational geometry and still be within inches of his original design. For example, a flat plane could be tangent to a cylinder and then tangent to a cone and still look like Frank's artistically twisted shapes. I noticed that by constructing the geometry in this way, I could get some repeatability out of the stone panel patterns.* » (Smith 2005, p.51).



conception, puis ces nouvelles surfaces furent retenues pour des raisons économiques (Smith 2005, p.53). En termes architecturologiques, on pourrait dire que les *échelles technique et économique*<sup>22</sup>, c'est-à-dire les opérations de conception relatives aux propriétés techniques et aux coûts du projet, étaient à l'origine peu opérantes dans le processus de conception. Elles deviennent prépondérantes par l'usage de la modélisation paramétrique. Quelques années plus tard, la rationalisation des formes est entrée dans les mœurs (Lindsey 2001, p.69) et est devenue par la suite une expertise de l'agence (Ehret 2009) et un sujet d'innovation technique, par exemple avec la mise en place de processus génératif des dessins de calepinage (cf. Figure 34 version F) (Shelden 2002, p.263-303).

Il était prévu que le découpage des pierres de la façade du WDCH ainsi que leur mise en œuvre soient effectués à l'aide de données extraites depuis le modèle paramétrique développé sur CATIA. L'entreprise chargée de ce lot travailla en forte collaboration avec Glymph et Smith (Smith 2005, p.53) et construisit un prototype de mur, exposé lors de la Biennale de Venise de 1993. Malgré la réalisation du prototype, la maîtrise d'ouvrage ne crut pas en la constructibilité du projet et décida de ne pas le poursuivre. Après le succès de la réalisation du Musée Guggenheim de Bilbao, en 1997 la maîtrise d'ouvrage et les assureurs décidèrent de relancer le projet. Le projet fut repris et modifié pour répondre aux nouvelles normes sismiques et, pour construire le parement de l'enveloppe extérieure en métal plutôt qu'en pierre (Glymph in Kolarevic 2005, p.110). Si à l'origine seule l'enveloppe extérieure avait été modélisée, lors de la poursuite du projet toutes les géométries ont été modélisées. En quelques années, d'autres opérateurs de CATIA ont rejoint Smith au sein de l'équipe C-Cubbed (cf. 3.2.2.1). Les modèles paramétriques développés pour le WDCH ont permis de représenter toutes les géométries et de contrôler leurs incohérences et leurs erreurs, particulièrement pour le contrôle des « clash », c'est à dire des intersections non voulues, entre les parois, les éléments de structure ou les flux. Cette représentation 3D a ensuite servi de support à la production des documents d'exécution.

### ***3.2.1.1.2 Appropriation de technologies industrielles : enjeux, limites, résistances***

Le travail de Smith et Glymph au début des années 90' fût de repérer des techniques pertinentes permettant de résoudre les difficultés rencontrées par Gehry Partners (comme représenter des surfaces courbes et les construire ou, numériser des objets physiques). Une fois ces techniques identifiées, il a fallu se les approprier et les intégrer à une logique d'agence fortement marquée par la tradition du « maître ».

Entre 1991 et 2003, de nouvelles méthodes techniques sont inventées au sein de Gehry Partners (Lindsey 2001, p.38). La modélisation de surfaces pour la rationalisation et la fabrication, ou encore le calepinage de surfaces courbes, sont les missions principales de Smith chez Gehry Partners (Smith 2005; Lindsey 2001). Pour Smith, faire entrer la modélisation paramétrique dans une agence d'architecture ressemblait à ces précédentes missions, cependant, la résistance qu'il a rencontrée a été inhabituellement forte.

---

<sup>22</sup> L'échelle *économique* correspond à l'activité de « concevoir une partie ou un tout de l'espace architectural en relation avec leur coût, ou en se livrant à des jeux d'échanges de coût entre diverses parties de cet espace architectural. » (Boudon et al. 2000, p.180).

Cette résistance provient de Frank Gehry lui-même<sup>23</sup>. En 2002, Frank O. Gehry explicite sa résistance face à la représentation numérique : « *It's very hard. I have in my head what I call a dream image, and then when I see it on the computer, it's dried out. So I have to hold the dream image while I am looking at the dried-out image, and it's excruciating. The computer takes the life out of it. I can only hold it so long. It's like putting your finger in a flame. Yes, it's exactly like that.* » (Gehry 2002).

La résistance provient également des collaborateurs de l'agence qui ont peur de l'obsolescence de leurs compétences (Smith 2005, p.41) ou qui souhaitent imiter le maître (Smith 2005, p.68). De plus, lors du processus de conception, la modélisation paramétrique permet de contrôler la cohérence de la géométrie représentée. Ce « contrôle » des géométries produites par d'autres collaborateurs est mal vécu dans un premier temps (Smith 2005, p.50). Par cette résistance, on observe que l'intégration de la modélisation paramétrique dans l'agence a un impact sur les activités des collaborateurs, même si ceux-ci utilisent toujours leurs mediums habituels.

S'il semble que la conception de modèles paramétriques soit, la plupart du temps, le fait d'experts (par exemples les membres de C-Cubbed cf. 3.2.2.1), il faut noter que certains architectes de Gehry Partners se sont investis également dans l'usage de CATIA pour le projet du musée Guggenheim de Bilbao par exemple (Smith 2005, p.72). Mais cet usage de la modélisation paramétrique semble relever (entre 1991 et 2003) d'une expertise stigmatisante au sein de l'agence. Cette catégorisation du concepteur semble préjudiciable pour le développement de sa carrière de chef de projet chargé de la conception architecturale. Ainsi Smith retrace le parcours de l'architecte Eva Sobesky qui, après un fort investissement dans l'usage de CATIA, décide de faire marche arrière. Smith écrit : « *She was concerned she might become pigeonholed as a CATIA operator within the office. She felt it would prevent her from progressing into a project management position and stated, —Architects did not work on the computers.* » (Smith 2005, p.75). La collaboratrice soulève ici l'impossibilité d'associer un rôle d'expert de la modélisation paramétrique (appelé « *CATIA operator* ») et un rôle d'acteur de la conception architecturale.

Le succès du musée Guggenheim de Bilbao fut un tournant majeur pour la reconnaissance de l'intérêt de l'usage de la modélisation paramétrique au sein de l'agence (Smith 2005, p.50).

### **3.2.1.1.3 Modèles paramétriques et Modèles physiques**

Une des premières tâches de Smith a été de modéliser un objet déjà conçu : le projet Fish. Pour ce premier travail de modélisation, des représentations en 2D ont été utilisées. Ces documents ont été produits par les collaborateurs à partir de la maquette aux moyens des méthodes du fil à plomb et de la projection lumineuse (Lindsey 2001, p.35). Par la suite, des méthodes spécifiques ont été élaborées en lien avec une technique de numérisation par un bras muni d'un pointeur de numérisation (le « *FARO arm* ») (Smith 2005, p.58). Trois méthodes de numérisation des maquettes physiques ont été établies : numériser des points situés au moyen de lignes horizontales régulièrement tracées sur l'objet, numériser les points limites de l'objet (extremums, arrêtes, *etc.*) ou encore numériser des points situés sur une

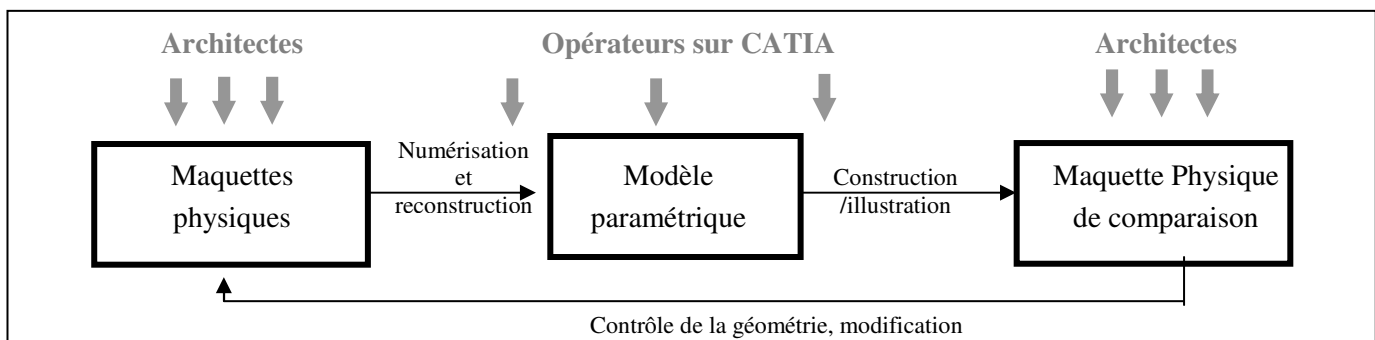
---

<sup>23</sup> « *Frank was very reluctant of the idea of using computers. He didn't trust them, thinking they would take the control of the design away from him.* » (Smith 2005, p.36).

grille superposée à l'objet (Lindsey 2001, p.65). Ces méthodes peuvent être combinées en fonction des caractéristiques de l'objet à numériser. Une fois les points numérisés, des courbes sont reconstruites puis des surfaces sont reconstituées. Cette reconstruction des courbes puis des surfaces implique des connaissances géométriques et techniques poussées (Lindsey 2001, p.35). Elle est souvent associée à des missions de rationalisation de la surface (cf. 3.2.1.1.4).

La numérisation peut parfois être une source d'inspiration pour la conception. Ainsi la forme de « tête de cheval » conçue pour l'entrée du projet de Résidence Lewis est issue de la numérisation d'un tissu épais rouge (Smith 2005, p.110). Cette forme sera réutilisée comme modèle pour le projet de la salle de conférence de la DG Bank de Berlin ainsi que pour une sculpture pour la Galerie Gagosian de Los Angeles. De la même façon, pour l'Experience Music Project, les architectes se sont inspirés des formes d'une guitare pour concevoir les différents bâtiments : de vrais fragments de guitares ont été utilisés pour la maquette physique, ceux-ci ont ensuite été directement numérisés, sans qu'il n'y ait d'interprétation des formes par de nouvelles maquettes (Smith 2005, p.94). La numérisation permet ici de réutiliser littéralement des « objets trouvés » et donc de s'inspirer de géométries existantes.

Ces allers-et-retours entre modèles physiques et modèles numériques dépendent en grande part de l'aversion de Frank Gehry pour la représentation informatique<sup>24</sup>. Ce dernier ne voulait pas évaluer la conception architecturale à partir d'un modèle numérique, des modèles physiques sont construits en parallèle des modèles paramétriques (Shelden 2002, p.74; Lindsey 2001, p.35). Ces maquettes, régulièrement tenues à jour, permettent aux architectes de prendre connaissance de la géométrie modélisée sur CATIA sans avoir à utiliser le modeleur (cf. Figure 33). Smith nomme ces maquettes des « modèles physiques de comparaison » (« physical comparison model ») (Smith 2005, p.55).



**Figure 33 : Liens et échanges entre maquettes physiques et modèles paramétriques**

Ces allers-retours entre maquettes physiques et maquettes numériques (cf. Figure 33) sont une façon d'enrichir le processus de conception en associant la précision de la modélisation paramétrique à la matérialité et à l'immédiateté de la maquette physique (Lindsey 2001, p.62). Pour Glymph, ce processus de matérialisation systématique des modèles paramétriques est très spécifique à la pratique de l'agence (Steele 2002, p.129).

Les différents modèles ne sont pas les supports des mêmes opérations. Glymph écrit: « *Modifications to form made in the computer were not made for the purpose of aesthetics- they were made for the purposes of a system fit. The aesthetic modifications were all made on physical model* »

<sup>24</sup> « Frank never looked at the computer screen to evaluate a design. He would only look at his physical design model. » (Smith 2005, p.55).

(Glymph in Kolarevic 2005, p.105). Cela exprime un cloisonnement des opérations de conception en fonction du médium : le modèle paramétrique permet le contrôle d'un système, principalement technique (cf.3.2.1.1.1) et le modèle physique permet les modifications « esthétiques », c'est-à-dire relatives à la manière dont le projet est vu, ce qui correspond à une *échelle optique*. Ce cloisonnement semble avant tout lié à l'organisation de l'agence et aux acteurs intervenant sur les différents médiums (cf. Figure 33).

#### **3.2.1.1.4 Modélisation paramétrique : rationalisation et fabrication**

Le modèleur CATIA a été choisi pour sa représentation en NURBS (Non-Uniform Rational Basis Splines) c'est-à-dire par équations polynomiales, qui permet une grande précision de la représentation des courbes et des surfaces (Lindsey 2001; Smith 2005; Steele 2002).

On peut dire que, sans CATIA, les formes conçues par Frank Gehry n'auraient pu être ni représentées ni construites dans des coûts et des délais raisonnables (Lindsey 2001; Kolarevic 2005; Smith 2011)<sup>25</sup>. L'usage de la modélisation paramétrique chez Gehry Partners permet de représenter des documents d'exécution impossibles à construire à la main. Cela s'illustre avec le cas de l'exécution du WDCH. Dans un premier temps, un architecte exécutif extérieur a été sollicité mais celui-ci ne disposant pas d'un savoir faire suffisant, n'a pu mener le travail à bien (Lindsey 2001, p.32)<sup>26</sup>. Les documents d'exécutions ont alors été refaits au sein de Gehry Partners.

La représentation en 3D permet une figuration précise (sans les approximations de l'épaisseur du crayon) et complète (dans la mesure d'une réduction de l'architecture en modèle d'architecture) de la géométrie du projet. Cette représentation permet de rassembler de nombreuses informations du projet par ailleurs en différentes représentations. Cette représentation permet de contrôler la géométrie conçue (par exemple un tuyau qui traverse un élément de structure) (Shelden 2002, p.57).

Le passage de la géométrie de la maquette physique à une géométrie paramétrique n'est pas anodin. La traduction des géométries en équations paramétriques oblige à une redéfinition de celles-ci, et une rationalisation des courbes non régulières (cf. Figure 34). L'enjeu est alors de rester au plus près des géométries conçues, en les rendant constructibles. La méthode de rationalisation choisie dépend des spécificités du matériau choisi pour la fabrication de la géométrie : la constructibilité par le métal est un enjeu récurrent de l'usage de la modélisation paramétrique chez Gehry Partners. L'enjeu pour Smith est de proposer des surfaces « metal-able » (Smith 2005, p.97). La rationalité des géométries met en jeu leur constructibilité mais aussi leur compréhension par les maîtres d'ouvrage, les entreprises ou encore les assurances. Glymph, affirme: « *In its general use, rationalization describes the process of converting the complex physical models, with all their indeterminacy, into a digital model where the complexity is tamed through mathematical description. It introduces "rules of constructability" into forms. This also has the effect of legitimizing the "crazy process"* » (Lindsey 2001, p.69). Lindsey associe ce travail de rationalisation à un enjeu de légitimation de la forme.

---

<sup>25</sup> Lindsey reprend une assertion de Zaera qui affirme: « *Many of the forms he is developing now are only possible through the computer [...] Bilbao is a perfect example. Prior to the computer applications in the office, they would have been considered something to move away from* » (Lindsey 2001, p.39).

<sup>26</sup> Gehry, à ce propos, affirme : « *We redid the working drawings completely from scratch because the former executive architect was not capable of doing them. You couldn't build our building from his drawing... They wanted to do it the way they always did.* » (Lindsey 2001, p.32).

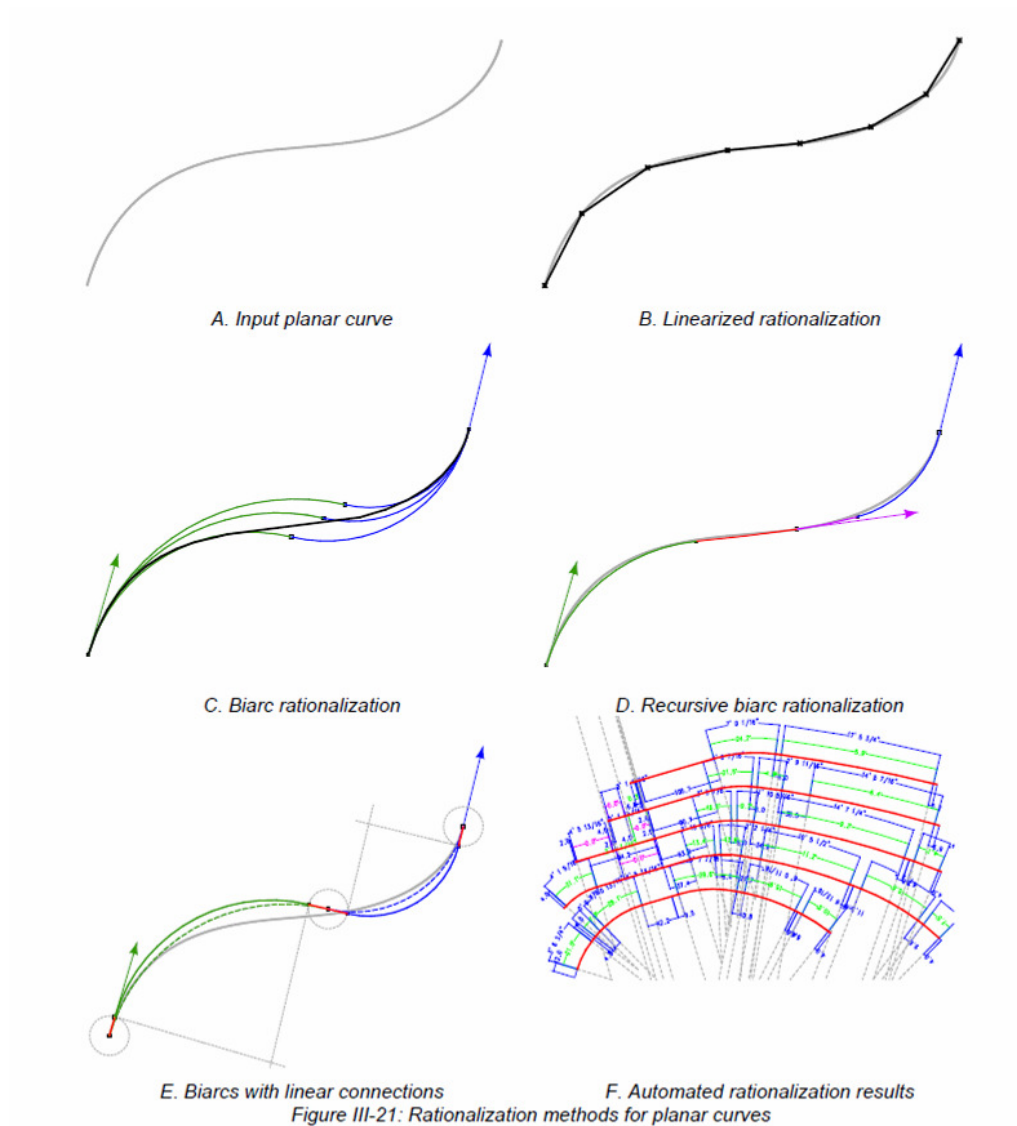


Figure III-21: Rationalization methods for planar curves

**Figure 34: cinq possibilités de rationalisation d'une courbe planaire par Dennis Shelden (Shelden 2002, p.84)**

Dans le même objectif de rendre constructible une architecture, certains composants (panneaux, éléments structurels, etc.) peuvent être rationalisés : ils sont ramenés le plus possibles à des formes semblables. Des composants standards sont moins onéreux à produire et à mettre en œuvre que des composants différents. L'enjeu pour Shelden est la maîtrise des coûts de fabrication des projets (Shelden 2002, p.43).

Au sein de Gehry Partners, la représentation paramétrique sert de support à l'évaluation des performances de l'architecture conçue (Shelden 2002, p.58). Le plus souvent, ces évaluations sont techniques (constructibilité d'une surface en métal (Smith 2005, p.97)) ou économiques (standardisation (Lindsey 2001, p.71), calcul de quantité de matière (Shelden 2002, p.33), etc.). Ces évaluations peuvent être effectuées en internes ou en externes lorsqu'il s'agit d'expertises techniques spécifiques (acoustiques, etc.). Le modèle paramétrique peut alors servir de support à l'évaluation si le transfert de données depuis CATIA vers les logiciels métiers utilisés est possible.

Le lien de la modélisation paramétrique à la fabrication est majeur chez Gehry Partners. Il a été présent dès le projet Fish. Néanmoins, ce recours à CATIA a tout d'abord été de l'ordre d'une aide à la représentation, plutôt qu'un lien direct du modèle à la production par commande numérique. En 1992, la fabrication de la peau du projet Fish par l'entreprise Permasteelisa a été menée en étroite collaboration avec Smith. Ce dernier a produit, à partir du modèle CATIA du projet, les patrons des bandes métalliques constituant l'enveloppe. Ces patrons sont les bandes développées imprimées sur papier à l'échelle 1, qui ont servi pour la découpe et la torsion des pièces (Lindsey 2001, p.38). Quelques années plus tard, des entreprises capables de produire des composants architecturaux directement à partir de machines à commandes numériques ont été développées.

Néanmoins, même quand les machines à commandes numériques ne sont pas utilisées, la production des documents d'exécution est une tâche pour laquelle la modélisation paramétrique est nécessaire. Ainsi pour le projet de Bilbao, 50 000 dessins et 60 000 heures de calcul ont été nécessaires pour produire les documents d'exécution de l'enveloppe extérieure (Lindsey 2001, p.44), alors que seuls les parements en pierres ont été produits par des machines à commandes numériques (Steele 2002, p.122). Pour Dennis Shelden, la fabrication est au centre de la conception des modèles paramétriques. Elle implique donc des méthodes et règles de constructions spécifiques (Shelden 2002, p.26)<sup>27</sup>.

### **3.2.1.1.5 Vers des « Master models » et des « Master Builders »**

La modélisation paramétrique d'un projet permet de regrouper toutes les informations nécessaires à sa description. Un tel modèle constitue ce que les acteurs de Gehry Partners appellent un « Master Model ». Les enjeux du *Master Model* sont multiples : il s'agit surtout de rassembler et contrôler les informations géométriques, générer les documents nécessaires à la fabrication, permettre le partage des informations entre différents acteurs (Shelden 2002, p.51). Le *Master Model* est une des pièces légales de description de l'ouvrage (Lindsey 2001, p.87). Originellement géométrique (Lindsey 2001, p.87), il évolue dans les années 2000 en intégrant également des informations sémantiques. La modélisation est alors nommée *Building Information Modeling* (BIM) (Shelden 2006).

Le « modèle unique » du Master Model, s'il résout des problèmes techniques, pose d'autres problèmes du côté des modeleurs. Dans le cas de CATIA, il dépend de la technologie de Dassault Systèmes qui évolue sans cesse. Smith note ainsi que, lors d'un projet à Düsseldorf, trois versions de CATIA ont été utilisées à cause des bugs rencontrés (Smith 2005, p.119). Le recours à un « modèle unique » pose également la question de la compatibilité des formats d'échange entre le modèle unique et les logiciels métiers des différents intervenants (pour l'évaluation structure, acoustique, thermique, etc.) (Lindsey 2001, p.75).

Finalement, le Master Model vise à faciliter : le partage d'informations : entre les architectes de la maîtrise d'œuvre (même si de 1991 à 2003 peu d'architectes manipulent les modèles paramétriques), entre les architectes et les BET (par exemple pour l'évaluation acoustique, les calculs de structure, etc.) et entre la maîtrise d'œuvre et les entreprises.

---

<sup>27</sup> Shelden affirme ainsi : « *a new set of guiding rules have been developed directly related to the materials and operations available to the processes of object making* » (Shelden 2002, p.26).

Les informations partagées par le Master Model sont à la fois géométriques et sémantiques. Le partage de données géométriques (comme des coordonnées de points par exemple) s'observe dès le projet Fish et sera analysé plus précisément dans le cas du modèle paramétrique du Pavillon Louis Vuitton (cf. 3.2.2). Un Master Model développé sur CATIA permet de partager des données sémantiques diverses : des noms de groupes ou d'objets, des attributs d'objets (natures, matériaux, etc.) (Shelden 2002, p.95), des couleurs, etc.

Le partage d'informations relatives à la fabrication *via* le Master Model, implique une réduction sensible du nombre de dessins d'exécution produits (Lindsey 2001, p.89). La production d'un Master Model conduit à l'implication de Gehry Partners dans le processus de production des documents d'exécution. Lindsey à ce propos reprend Cocke qui affirme : « *Both time and money can be eliminated from the construction process by shifting the design responsibility forward.* » (Lindsey 2001, p.89). On observe ici que l'intégration de la modélisation paramétrique chez Gehry Partners transforme le processus de conception, ici en permettant d'approfondir la définition du projet et son *embrayage*<sup>28</sup> sur le réel.

L'utilisation d'un Master Model implique une nouvelle mission : celle d'organiser les accès de chacun au modèle, de gérer et contrôler le modèle. Cette mission est remplie par un acteur nommé par Gehry Partners le « Master Builder » (Lindsey 2001, p.89). Dans d'autres contextes mais dans une acception très proche il est aussi nommé « model manager » (Bonora 2012). Ce rôle est important car il implique une responsabilité autour du partage des données, entre autres à propos du contrôle de l'actualité des géométries, de leur cohérence et de leurs échanges avec les collaborateurs. Pour Stéphane Hanrot ce rôle a une fonction pivot, qui se distingue de la mission de la maîtrise d'œuvre (Hanrot in Terrin 2005, p.60). Chez Gehry Partners de 1991 à 2003, cette pratique est assumée par la maîtrise d'œuvre en collaboration avec l'équipe de spécialiste « C-Cubbed » fondée par Rick Smith. Il sera développé comme un service à part entière au sein de l'entreprise Gehry Technologies.

Aujourd'hui le Master Model est non seulement un document légal intégré à la description du projet, mais également le document faisant foi en cas d'informations contradictoires (Lindsey 2001, p.89). Cette logique de document unique ne s'inscrit pas encore pleinement dans la pratique usuelle des maîtres d'œuvre, BET et entreprises, qui redessinent les projets (Shelden 2002, p.33). La transmission des informations est contrôlée et peut poser des difficultés de coordination pour des projets aux géométries complexes (Shelden 2002, p.41). En conséquence, il est arrivé à Gehry Partners de créer des contrats privilégiés avec certaines entreprises hautement qualifiées : l'entreprise pouvant amener son expertise à Gehry Partners dans la définition de certains détails techniques, et Gehry Partners de son côté pouvant produire certaines représentations nécessaires à l'entreprise (Shelden 2002, p.42). Quelque soit les échanges d'informations opérées, le « Master Model » reste néanmoins « *the official format for 3D data on the firm's project* » (Shelden 2002, p.26).

Nous venons de pointer les principaux tenants et aboutissants de l'intégration de la modélisation paramétrique chez Gehry Partners, en particulier en analysant les missions pour lesquelles elle est utilisée. Nous avons vu que la modélisation paramétrique est indispensable pour la représentation et la fabrication des projets, mais qu'elle est mise en œuvre en fonction de tâches déterminées par des acteurs

---

<sup>28</sup> En architecturologie, le concept d'embrayage désigne « la prégnance de l'espace réel, à la fois renvoi obligé et finalité » dans la conception architecturale (Boudon et al. 2000, p.98)

qui ne sont pas ceux de la conception (cf. Figure 33). L'usage de la modélisation paramétrique répond à des enjeux économiques et techniques et transforme le processus en l'élargissant à l'exécution. Dans le paragraphe suivant, nous nous intéressons aux effets de la modélisation paramétrique sur la conception architecturale.

### **3.2.1.2 Les effets de la modélisation paramétrique sur la conception architecturale**

Des études sur le travail de Frank Gehry montrent qu'il existe des récurrences dans son processus de conception. Ainsi la plupart des projets commencent par une phase d'exploration du programme (Lindsey 2001, p.50; Lecourtois 2004) au moyen de croquis et de maquettes physiques, supports de la conception de l'organisation du programme sur le site. Cette phase de conception est un moment d'échange et de collaboration avec la maîtrise d'ouvrage. Frank Gehry produit des croquis qui sont comme des « images » (Gehry 2002). Pour Lindsey, ces images ont une force évocatrice pour le projet. Elles sont ouvertes à de multiples interprétations mentales (Lindsey 2001, p.53). Ces interprétations permettent l'exploration du projet, tandis que l'image numérique selon Gehry ne posséderait qu'une interprétation fermée, sans force évocatrice (Gehry 2002). Ces croquis sont interprétés par ses collaborateurs les plus proches en maquettes physiques (Lindsey 2001, p.55). Ces maquettes physiques sont le support de la production de documentation 2D du bâtiment et de sa représentation numérique en 3D *via* CATIA. Frank Gehry revendique l'usage du dessin comme média de la conception, il associe sa pensée à l'usage du croquis : « *I think that way* » (Lindsey 2001, p.53).

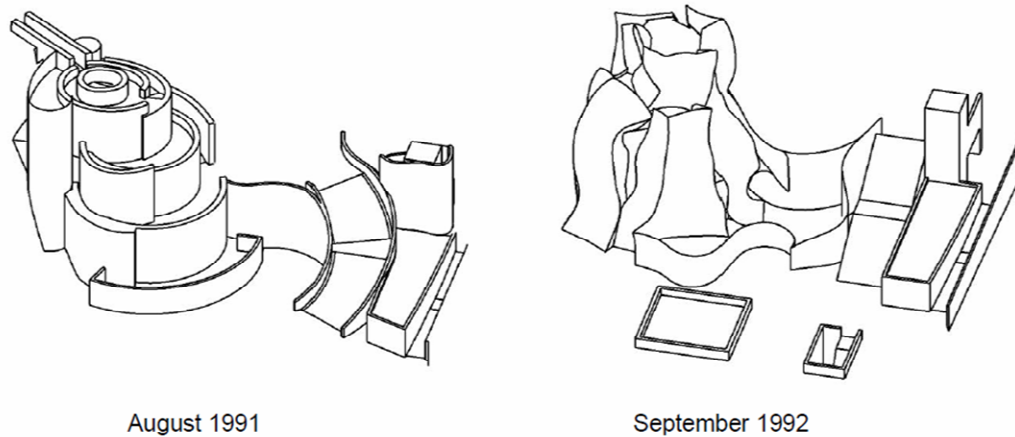
Mais si la modélisation paramétrique n'intervient que pour des missions délimitées (représentation, fabrication, partage de données), le croquis et la maquette physique étant le medium favori de Frank Gehry et des collaborateurs de l'agence, ce n'est pas pour autant que l'on peut en conclure qu'elle n'intervient pas dans la conception. Dès 2001, Lindsey affirme « *Despite the efforts to the contrary, the introduction of digital tools did change Gehry's process.* » (Lindsey 2001, p.49).

#### **3.2.1.2.1 Une transformation de l'espace de référence : rôle de l'appropriation de l'outil**

Nous l'avons vu précédemment, le recours à la représentation et à la fabrication par l'usage de CATIA a permis à l'agence de construire de nouvelles formes complexes. Pour la plupart des acteurs de la modélisation paramétrique, la conscience de cette potentialité aurait, en soi, transformé la conception de Frank Gehry (Lindsey 2001; Kolarevic 2005). Frank Gehry revient sur sa propre pratique avec un discours très proche: « *I started making shapes that were hard to draw. That led us to the computer and to Catia software which made me realize the possibilities and the level and degree of accuracy you could create in your documents and your relationships because of the software.* » (Gehry 2007). Smith illustre cette évolution par l'exemple du cas d'une partie du WDCH : « the founder room », dont il donne des schématisations correspondant à différentes versions du projet dans le temps (cf. Figure 35, extrait de Smith 2005, p.54). On observe dans ces formes une nette évolution des géométries, constituées de volumes euclidiens en août 1991 (en particulier des portions de cylindres), les géométries



prennent des courbures beaucoup plus variées en septembre 1992 (cf. Figure 35, extrait de Smith 2005, p.54).



**Figure 35: Schématisations de deux versions de la "chambre des fondateurs" du projet du WDC de Gehry Partners, extraites de *Changing the shape of architecture digitally* de Rick Smith (Smith 2005, p.57)**

Tout se passe comme si l'imaginaire construit par Frank Gehry autour des potentialités de la modélisation paramétrique, en particulier sur ce qu'il est possible de représenter et de fabriquer, faisait émerger des idées, des réflexions, voire de nouveaux *espaces de référence* (Boudon et al. 2000, p.111) opérant dans de nouvelles mesures architecturales.

### **3.2.1.2 Enjeux et maîtrise de la fabrication : transformation des pertinences et des unités de conception**

Les enjeux de la modélisation paramétrique pour la fabrication sont majeurs. La prise en compte de la construction transforme les *pertinences* et les *échelles* sollicitées. Ainsi, en 1992, l'usage de la modélisation paramétrique pour la représentation des surfaces du Fish permet l'évaluation de celles-ci, puis leur rationalisation (cf. 3.2.1.1.1). Sans la modélisation paramétrique, évaluation et rationalisation ne pouvaient être menées par Gehry Partners. Les *échelles économique et technique*, jusque là peu sollicitées par les équipes de conception, sont facilement mises en œuvre dans ce cadre de la conception du modèle paramétrique.

Dans le cadre de la conception du projet de Centre Ray et Maria Stata pour le MIT, le collaborateur Marc Salette affirme: « *On MIT where the 3D model was a tool almost from the beginning, there can be a conversation about how the building is going to be put together in terms of detailing, but also in terms of construction sequence, much earlier in the process, because the understanding of what we're trying to achieve technically and architecturally is starting to take place a lot earlier.* » [arcspace.com]. Salette rend compte ici d'une transformation des processus de conception par la prise en compte précoce des enjeux de construction (détails techniques, chantier).

Les points de vue technique et économique ont pris, grâce à la modélisation paramétrique, une toute autre ampleur : l'architecte, au moyen de la modélisation paramétrique, se retrouve investi d'une mission beaucoup plus large, liée à l'exécution, à la création et au maintien du modèle paramétrique. Ainsi pour Lindsey, la limite des compétences et des missions de l'architecte est repoussée (Lindsey 2001, p.87).

### **3.2.2 Le cas de Gehry Technologies Europ et Gehry Partners pour la fondation Louis Vuitton pour la Création**

Dans le cas du Pavillon de la Fondation Louis Vuitton pour la Création, nous nous intéressons spécifiquement à la mission de l'équipe Gehry Technologies Europ.

#### **3.2.2.1 Formation de l'entité "Gehry Technologies"**

A la fin de la période d'essai de Smith, en 1993, il fût décidé que celui-ci garderait un statut de consultant indépendant, à la fois à cause de la résistance à la technologie qui existait alors dans l'agence, ainsi que pour pouvoir utiliser la modélisation sur CATIA plus largement qu'uniquement pour les missions internes à l'agence (Lindsey 2001, p.87). Dès 1993, Gehry et Glymph ont voulu utiliser la modélisation paramétrique et la diffuser pour rentabiliser leur investissement. C-Cubbed collabora surtout avec les entreprises directement impliqués dans les projets de Gehry Partners (Lindsey 2001, p.87). L'équipe de C-Cubed était formée d'opérateurs CATIA, majoritairement ingénieurs de formation, spécialisés dans la modélisation paramétrique, et non des architectes (Smith 2011).

Aujourd'hui, l'expertise de modélisation paramétrique de Gehry Partners est poursuivie par l'entreprise Gehry Technologies. Les acteurs de Gehry Technologies ne sont pas ceux de C-Cubbed : l'ampleur et les objectifs de l'entreprise sont différents (Smith 2005, p.155)<sup>29</sup>. Aujourd'hui, l'équipe experte en modélisation paramétrique « Gehry Technologies », possède de nombreux bureaux<sup>30</sup>. Depuis 2003, les équipes de Gehry Technologies travaillent comme expert de la modélisation paramétrique pour divers architectes concurrents de Gehry Partners, comme Ateliers Jean Nouvel, Zaha Hadid Architects ou Herzog et Demeuron (Ehret 2009).

En 2003, un agrément est négocié entre Gehry Technologies et Dassault Systèmes : Gehry Technologies devient un partenaire commercial, autorisé à vendre CATIA V5. Une version métier pour architecte de CATIA est développée : c'est « Digital Project » (cf. 2.3.3). Aujourd'hui Gehry Technologies développent d'autres outils<sup>31</sup>.

---

<sup>29</sup> En 2003, Rick Smith ainsi que la totalité des membres de C-Cubbed sont remerciés (Smith 2005, p.155). Glymph et Shelden fondent alors l'entreprise « Gehry Technologies ». En 2006, Jim Glymph quitte Gehry Partners et Gehry Technologies (Smith 2005, p.158). Seul Dennis Shelden (intégré à Gehry Partners en 1997) poursuit l'expérience développée.

<sup>30</sup> Le bureau principal (« Gehry Technologies Global ») est à Los Angeles dans les locaux de Gehry Partners. Un bureau (« Gehry Technologies Europe ») à été ouvert à Paris en 2008 pour le projet de la Fondation Louis Vuitton pour la Création. Il existe également des bureaux à New York, Abu Dhabi, Shanghai, Guangzhou, Beijing, Séoul et Sydney [gehrytechnologies.com].

<sup>31</sup> Ces outils sont par exemple comme le système de partage de données GTeam [gteam.com] ou encore la plateforme Euclid (Witt 2012)

Les missions de Gehry Technologies sont multiples. Nous nous intéresserons ici plus particulièrement à l'intervention de Gehry Technologies sur deux projets : le Pavillon Louis Vuitton pour la Création de Gehry Partners (3.2.2) et le Qatar National Museum des Ateliers Jean Nouvel (3.2.3). Ces deux cas seront abordés sous l'angle de la relation entre modélisation paramétrique et conception architecturale, en particulier au travers de l'analyse des relations des équipes de conception et des membres de Gehry Technologies.

### **3.2.2.2 Le cas du pavillon de la Fondation Louis Vuitton pour la Création : description du contexte**

Les premières esquisses du Pavillon de la Fondation Louis Vuitton pour la Création, conçu par Gehry Partners, datent de 2006. Le chantier du bâtiment a commencé en 2009 au Jardin d'Acclimatation à Paris. Pour la phase d'exécution, plus de trois cents professionnels différents et dispersés interviennent simultanément. Ainsi, les architectes sont à Los Angeles, les ingénieurs façade, structure et fluide à Paris, et, d'autres intervenants sont en Allemagne, en Italie et en Angleterre. Cette situation pose la question des conditions de partage du travail et de la communication des informations.

Pour favoriser la cohérence du travail, les intervenants français (les Bureaux d'Etudes Techniques (BET) Setec, RFR, TESS, etc. et l'architecte local de l'agence Studios Architecture) ont été rassemblés sur un même site. Ils partagent les mêmes locaux et les mêmes ressources informatiques. La mise en place de ce plateau à Paris permet, selon le maître d'ouvrage (le groupe LVMH), de favoriser la communication entre intervenants. Pour optimiser et rentabiliser le travail des intervenants, le maître d'ouvrage a imposé, à tous, l'usage du logiciel utilisé par Gehry Partners : Digital Project.

Gehry Technologies, société de développement de Digital Project, a fondé le bureau Gehry Technologies Europ à Paris en 2007-2008, afin de former les intervenants du Pavillon Louis Vuitton à Digital Project. Elle les a également assistés dans la conception du modèle paramétrique commun (Witt, 2009). La plupart des ingénieurs sur le site de Paris note la difficulté éprouvée à changer leurs habitudes de travail et à apprendre un nouveau logiciel. De nombreux efforts ont été fournis pour favoriser la collaboration sur ce projet, tant du côté de la maîtrise d'ouvrage que de la maîtrise d'œuvre.

Malgré ces stratégies de collaboration (réunion des intervenants sur un même lieu physique et une même plateforme : le modèleur Digital Project), deux modèles paramétriques distincts ont été développés : l'un par les architectes de Gehry Partners à Los Angeles et l'autre par les équipes de Paris.

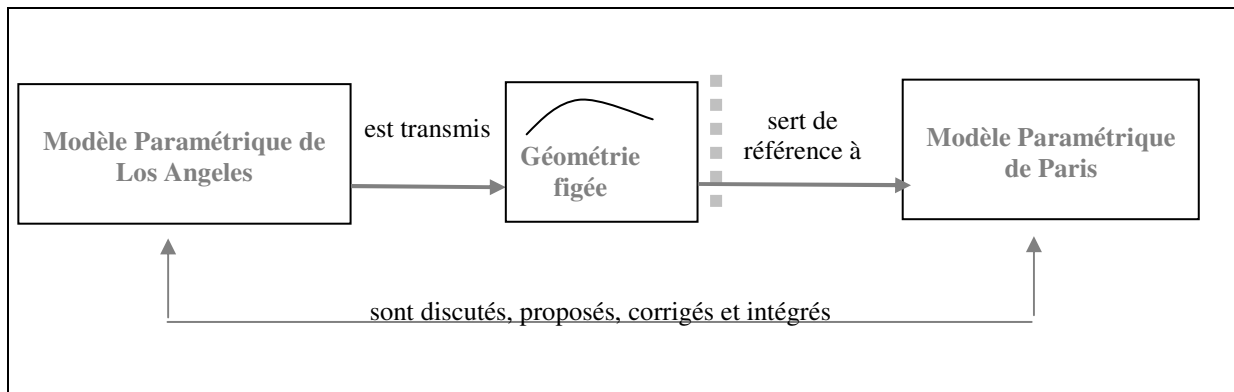
Le modèle développé par Gehry Partners n'est pas partagé avec les ingénieurs. Une rupture franche est établie entre les deux modèles : le travail des équipes de Paris ne vise pas à compléter directement le modèle de Gehry Partners mais à créer un nouveau modèle (cf. Figure 36). Le modèle de Gehry Partners est communiqué aux équipes de Paris sous la forme d'une « géométrie morte »<sup>32</sup>, non paramétrique. Cette géométrie ne comporte plus la mémoire de sa conception mais a été « sauvée comme un résultat »<sup>33</sup>. Cette géométrie conserve l'organisation et les nominations attribuées aux différentes données. Si l'utilisation d'un même modèleur permet d'échanger des données compatibles, le

---

<sup>32</sup> Ce vocabulaire correspond à une terminologie propre aux fonctionnalités de Digital Project.

<sup>33</sup> *Idem.*

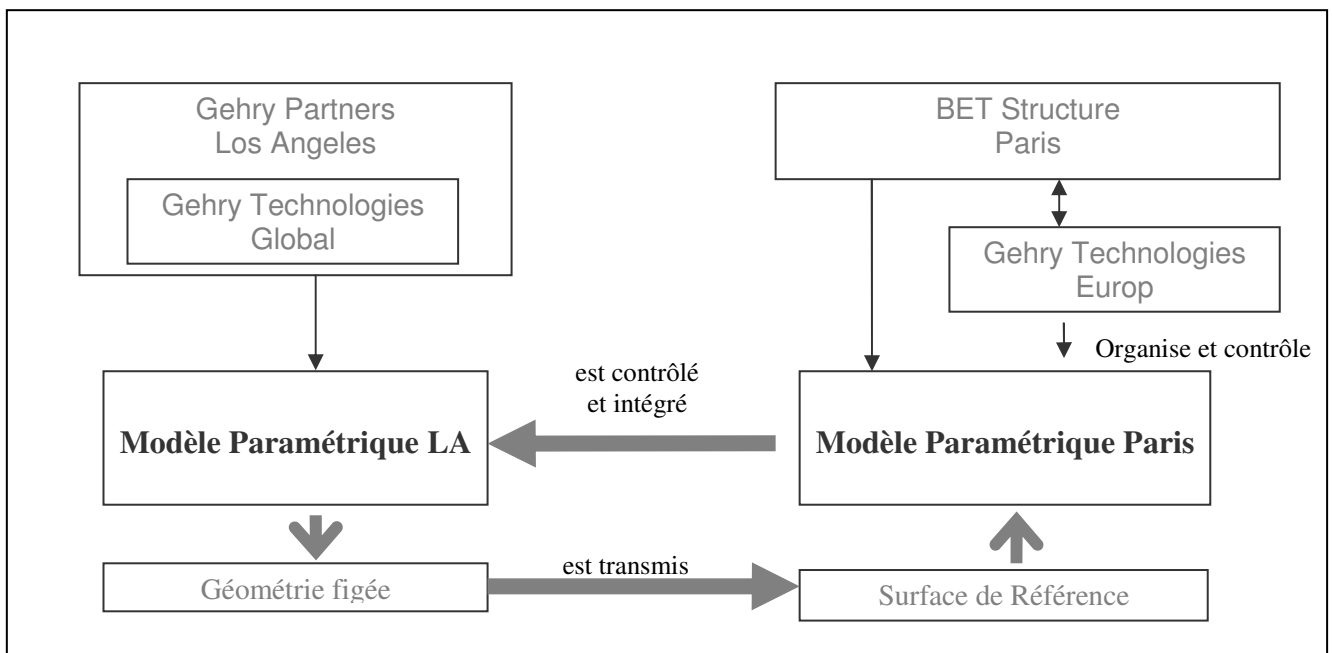
cloisonnement établi entre le modèle de Gehry Partners et le modèle de Paris pose question (cf. Figure 36).



**Figure 36: Partage de l'information grâce à la modélisation paramétrique pour le cas du pavillon Louis Vuitton**

Un nouveau travail de conception est fait par les équipes de Paris. Il porte sur la conception de la structure, des détails d'exécution, etc. Il s'appuie sur la « géométrie morte » communiquée par Gehry Partners et implique une nouvelle conception du modèle paramétrique.

Les liens entre le modèle de Los Angeles et celui de Paris sont contrôlés et ponctuels. Le modèle figé transmis par Gehry Partners est utilisé comme géométrie de référence pour le modèle des équipes de Paris. Les mises à jour sont effectuées sous contrôle : un changement même minime de la géométrie de référence peut avoir d'importantes répercussions sur le modèle de Paris. Par ailleurs, le modèle de Paris est « nettoyé » en fin de semaine, c'est-à-dire retravaillé, dans le but d'être communiqué aux équipes de Gehry Partners (cf. Figure 37). Cette communication semble permettre à l'équipe de Los Angeles de préciser son propre modèle.



**Figure 37: Contrôle et communication des données entre BET et Architectes sur le Pavillon de la Fondation Louis Vuitton pour la Création**

### 3.2.2.3 Opération de conception du modèle paramétrique de Paris : le cas du découpage

Dans le modèle général, on observe deux découpages différents qui correspondent à des temps différents : ce sont le découpage par lots et le découpage par « nœuds ».

#### - Découpage par lots

Ce découpage correspond à la répartition des tâches entre les différentes équipes. Il est lié à l'organisation traditionnelle du chantier. On distingue ainsi le lot « architecture et finition » qui appartient à l'architecte local de l'agence Studios Architecture, le lot « gros œuvre » de SETEC ou encore, le lot « parois vitrées » de RFR, etc. Ce découpage répond à une pertinence liée à l'organisation du travail. Il permet d'identifier les objets géométriques sur lesquels chaque équipe travaille spécifiquement. Ce découpage est associé à des droits d'écriture qui dépendent de la hiérarchie des intervenants. Il crée un cloisonnement de la conception entre les sous-modèles sur lesquels travaillent les différentes équipes.

Si ce découpage semble avant tout fonctionnel, il répond également à des distinctions opérées par Gehry Partners. Dans la « géométrie morte » transmise à Paris, on retrouve un système de nominations qui reflète des intentions de conception. Ces nominations distinguent des éléments contrastés : ainsi les « voiles » désignent une partie vitrée, et l'« iceberg » un « bâtiment dans le bâtiment » (Gehry, 2010) pour les espaces d'exposition. Ces partis architecturaux sont lisibles dans le choix des matériaux et du traitement des éléments (vitrage, béton, etc.) et se retrouvent, de fait, dans le découpage des ouvrages... Ainsi le découpage de l'« iceberg » correspond au gros œuvre et celui des « voiles » aux verrières. On retrouve ici une organisation de l'arbre des spécifications de Digital Project décrit ainsi par Andrew Witt : « l'arborescence permet d'organiser et de hiérarchiser le modèle : le projet lui-même est soumis à une certaine charte d'organisation, celle-ci peut naturellement s'appliquer à la structure du projet afin de modulariser le travail associé » (Witt et al. 2009) Selon cet auteur, la structure du modèle paramétrique guiderait à la fois le projet et l'organisation du travail en commun.

#### - Découpage par « nœuds »

Ce découpage porte sur les zones où la rencontre de lots est particulièrement complexe. Contrairement au découpage précédent qui était composé de « products » et qui correspondait à la structure même du modèle, ce découpage est surajouté. Il se fait par script et crée un nouveau « product ». Ces découpages, appelés « section box » par Gehry Technologies (Ehret 2009), sont au nombre de quatre-vingt environ. Ils sont mis en œuvre lors de réunions régulières menées par Gehry Technologies et rassemblent les équipes concernées par les ouvrages en cause. Ces réunions visent à vérifier la cohérence du modèle et à coordonner les zones sensibles. Cela permet par exemple de contrôler et de partager les informations entre le travail de Setec et celui de RFR, sur les points d'appuis de la structure des verrières sur les ouvrages en béton. Ces reprises de charges sont gérées par des éléments architecturaux spécifiques: les « tripodes » (Figure 38). La modification d'un tripode par l'équipe chargée des verrières (RFR) doit immédiatement être transmise à l'équipe qui dessine et calcule le gros œuvre (Setec). Tandis que le précédent découpage cloisonnait les objets de chaque

équipe, ce second découpage permet un contrôle de la cohérence du travail entre les équipes et un partage des informations.



**Figure 38 : Maquette du pavillon Louis Vuitton pour la Création. Sources: Joewen1980, sous licence CC By NC ND 2.0**

Ces deux découpages du modèle sont donc liés à l'organisation du travail et à la répartition des tâches entre les différents intervenants. A ce niveau général d'observation, le modèle semble avoir été construit pour optimiser la communication et le partage du travail post-conception.

### **3.2.2.4 Partage de données au sein du modèle paramétrique de Paris**

Au sein du modèle de Paris, des données transversales à différents lots sont partagées. Ces données sont en particulier des points de référence, repères définis par le géomètre mais aussi points de référence structurels : par exemple marquant l'ancrage d'un élément de structure primaire sur un des « icebergs » de béton. Ce partage permet d'assurer la cohérence de la géométrie du modèle. Dans ces cas, le partage de ces données est automatisé par l'utilisation de « publication ». Ce système de partage propre à CATIA permet d'explicitement, au sein d'un modèle partagé, les données qu'un opérateur « rend publiques » : c'est-à-dire les données qu'il identifie comme étant pertinentes et utilisables par son collaborateur et bien définies par rapport à l'avancée de son modèle (Mineur 2011).

De même, les différents concepteurs du modèle paramétrique de Paris partagent les « surfaces de référence » communiquées par l'agence Gehry Partners (cf. Figure 37). La modification du modèle de Los Angeles par Gehry Partners entraîne une mise à jour globale du modèle de Paris, même si cette mise à jour est ponctuelle. En effet, le lien entre le modèle de Paris et le modèle de Los Angeles n'est pas immédiat puisque le modèle de Paris est construit à partir des « géométries mortes » de Gehry Partners (Figure 37).

Dans le cas du Pavillon de la Fondation Louis Vuitton, les équipes rassemblées sur le site de Paris partagent donc un même modèle paramétrique. Le découpage de ce modèle est tel que chaque équipe travaille sur des objets géométriques spécifiques et cloisonnés. Notre analyse permet de montrer que la modélisation paramétrique est un support de collaboration mais ne permet pas d'affirmer que l'objet de cette collaboration est la conception.

Digital Project a assisté les équipes de Paris pour penser les liens entre les travaux des différents intervenants. Il assiste la cohérence globale du modèle, grâce aux « publications » et aux « surfaces de référence », même si cette automatisation reste contrôlée par une équipe spécifiquement dédiée à la coordination du travail des intervenants.

### 3.2.3 Le cas de Gehry Technologies et AJN pour le projet Qatar National Museum (QNM)

Les premiers échanges entre les Ateliers Jean Nouvel (AJN) et Gehry Technologies Europ ont lieu en 2008 en vue d'intégrer Digital Project à l'agence (Kubota 2012, p.34). Chez AJN, le projet du Musée pour la culture qatarie, ou Qatar National Museum (QNM) commencé en 2005, trouve sa forme actuelle en 2008 (Kubota 2012, p.38). C'est un projet symbolique pour l'état qatari. Premier monument visible en arrivant depuis l'aéroport, le musée « doit refléter à la fois le passé et le futur de l'émirat » (Kubota 2012, p.37).



Figure 39 : Vue du Qatar National Museum d'Ateliers Jean Nouvel. Sources: [jeannouvel.com]

Le projet fut développé dans un premier temps, au sein d'AJN, sur Rhinocéros. La modélisation 3D a permis alors de « vérifier et améliorer » les formes, mais aussi d'extraire des représentations 2D (Kubota 2012, p.38). Les limites de ce travail sur Rhinocéros sont apparues lorsque le projet est devenu plus détaillé, lors de la phase PRO. Les principales difficultés de ce travail sur Rhinocéros étaient liées au poids excessif du modèle. Malgré différentes stratégies (réduire les géométries représentées, ne pas représenter les intersections, utiliser les machines les plus puissantes du marché) le modèle était difficilement maniable. L'équipe de Gehry Technologies sur Digital Project a été choisie pour intervenir sur le projet.

L'usage de Digital Project a permis à la fois d'évaluer et de rationaliser les géométries conçues, et de rassembler et d'organiser les informations relatives au projet.

L'usage de Digital Project a permis le calepinage des bâtiments en forme de disques, l'optimisation du vitrage pour la pose des menuiseries et le contrôle des espaces minimum entre les disques pour la mise en œuvre du calepinage (Kubota 2012, p.39). On retrouve ici les enjeux de rationalisation et

d'évaluation spécifiques à l'usage de la modélisation paramétrique chez Gehry Partners (cf. 3.2.1). Ils sont ici liés à la mise en œuvre du chantier.

L'expertise de Gehry Technologies a permis d'organiser la représentation 2D du projet (par la mise en place de systèmes de calques et d'étiquettes), d'automatiser le métré du chiffrage, et de mettre au point une procédure de contrôle de la géométrie pendant le chantier (Kubota 2012, p.39).

L'intervention de Gehry Technologies s'est faite en quatre temps (Kubota 2012, p.41):

- une phase de formation de trois membres d'AJN sur Digital Project,
- une phase de travail en « équipe mixte », comprenant des membres de GT ainsi que des membres d'AJN. Cette équipe fût chargée de la conception d'un modèle paramétrique du projet à partir de la géométrie précédemment représentée au moyen du modèleur Rhinocéros ;
- une phase d'implémentation du modèle paramétrique : la géométrie fut affinée via la collaboration avec les BET et par le travail de certains sujets non résolus jusqu'alors (calepinage des disques, plafonds). Le modèle paramétrique fût également implémentée en termes de méthodes (organisation de l'arbre des spécifications, mise en place d'un protocole d'extraction des représentations 2D, mise en place de la génération du métré, etc.) ;
- la dernière phase est une phase courte de production de représentations 2D et de données finales au moyen du modèle.

Pour cette mission, une équipe mixte de sept personnes mêlant des membres d'AJN, des experts de la conception architecturale formés à Digital Project, et des membres de Gehry Technologies, experts de la modélisation paramétrique, a été constituée. L'équipe était implantée et opérait au sein même d'AJN. Une porosité entre les missions de conception et de modélisation semble s'être installée. Les différents acteurs ont partagé leurs expertises<sup>34</sup>. Les missions de modélisation et certaines missions de conception, se sont petit à petit croisées (Occhipinti 2011). On peut parler ici d'hybridation des pratiques, par le croisement de plusieurs compétences donnant lieu à un nouveau processus de conception. L'hybridation opère au sein de l'équipe et elle procède d'un échange de compétences au sein d'une expérience commune.

D'après Stéphane Hanrot, la collaboration entre acteurs est une organisation de travail récurrente dans les cas d'usage des technologies numériques. Lors d'une étude menée entre 1999 et 2001 sur les nouvelles pratiques de projet, liées aux technologies numériques, il esquisse quelques dynamiques spécifiques à l'évolution de l'organisation des acteurs de la maîtrise d'œuvre. En particulier, il soulève que les deux organisations types de la maîtrise d'œuvre : *organisation hiérarchique*, autour d'un individu pivot et *organisation coopérative*, avec des acteurs en interaction sans nivellement hiérarchique (Hanrot in Terrin 2005, p.52 - 54), sont en court d'hybridation. En 2005, Hanrot pointe une transformation d'une partie de l'organisation du projet et de certaines phases de projets. Pour Hanrot,

---

<sup>34</sup> Edmondo Occhipinti, responsable de cette mission chez Gehry Technologies, exprime l'échange de compétences sous forme de boutade : pour lui les membres de Gehry Technologies ont formé les architectes d'AJN à Digital Project, et les architectes d'AJN ont « formés les membres de Gehry Technologies à Jean Nouvel » (Occhipinti 2011).



sous la pression de l'évolution des TIC entre autres facteurs<sup>35</sup>, l'organisation des acteurs et des pratiques pourrait s'hybrider, et mettre en cause la pratique de la conception par un individu pivot (Hanrot in Terrin 2005, p.62). Cette pratique de la conception du projet mené par un groupe d'individus est celle que nous avons observée dans notre cas.

---

<sup>35</sup> « Sous la pression conjuguée de l'évolution des TIC et du cadre juridique, le modèle d'organisation hiérarchique des acteurs de la maîtrise d'œuvre qui prévaut devrait s'amender et intégrer des organisations coopératives. » (Hanrot in Terrin 2005, p.62).

### **3.3 Pratiques de consultation interne : le cas de Foster and Partners**

#### **3.3.1 SMG et ARD Team chez Foster and Partners: in-house consultancy**

L'agence Foster and Partners compte en 2012 plus d'un millier d'architectes. Ces architectes sont répartis en six équipes de conception, appelées « Design Teams », d'environ 150 membres chacune. Cette taille est revendiquée comme un atout permettant de gérer et de mener à bien des projets complexes [fosterandpartners.com]. Au sein de l'agence, des équipes spécialisées dans divers domaines (projet urbain, développement durable, *etc.*) appuient les Design Teams (D'Alligna 2009) sur le mode de la consultation interne, dite « in-house consultancy » [fosterandpartners.com]. Nous nous intéressons ici en particulier aux équipes « Applied Research and Development » (ARD), « Specialist Modelling Group » (SMG) et « Specialist Modelling Group Environmental » (SMGE), spécialisées dans l'usage d'outils numériques parmi lesquels se trouve la modélisation paramétrique.

Ces trois équipes sont issues du succès du SMG fondé en 1998 par Hugh Whitehead, un des Partners de Norman Foster, pour le projet du City Hall de Londres. Les activités du SMG rencontrent un franc succès au sein de l'agence. Cinq ans après sa création, en 2003, le SMG a déjà été impliqué dans soixante trois projets et l'équipe comprend quatre membres pour assister quatre cents architectes (Whitehead in Kolarevic 2005, p.83). A la fin des années 2000, les effectifs de l'équipe ont doublé : le SMG compte dix personnes pour assister plus de mille architectes (Whitehead 2009b). En 2011, Le SMG est divisé en trois équipes : l'équipe « Applied Research and Development » (ARD), le « Specialist Modelling Group Environmental » (SMGE) et le « Specialist Modelling Group » (SMG) (F. Aish & Chronis 2012).

Le panel des missions de ces équipes est large. Ces groupes interviennent principalement comme experts en géométrie et en informatique (en modélisation autant qu'en développement d'outils), mais assistent également des procédures de collaboration et de fabrication assistées par ordinateur, ainsi que la qualité environnementale des projets. Le SMG poursuit une mission de modélisation de projet. L'ARD développe des programmes de plus bas niveaux (c'est-à-dire que l'équipe développe des codes plutôt que des scripts) et vise une pratique de recherche appliquée (F. Aish & Chronis 2012). Le SMGE, quant à lui, traite tout particulièrement des enjeux de performances environnementales. Le SMG est dirigé par Hugh Whitehead, le SMGE par Irène Gallou et l'équipe ARD par Francis Aish.

Les pratiques du SMGE ne sont pas abordées dans cette recherche. Les pratiques du SMG (depuis 1998) en revanche sont particulièrement étudiées ici.

La programmation de modèles paramétriques et de scripts est une part majeure de l'activité du Specialist Modelling Group. C'est l'activité qui nous intéresse le plus dans notre étude. Elle ne peut cependant être isolée des autres missions du SMG que sont, entre autres : organiser les liens entre équipes de conception et consultants (internes et externes), organiser la collaboration au sein d'une

même équipe au travers d'une bonne propagation des transformations dans les fichiers décrivant le projet, faire de la veille technologique ou, pour l'ARD, de la Recherche et du Développement<sup>36</sup>.

Pour comprendre l'usage de la modélisation paramétrique par le SMG et l'ARD au sein de Foster and Partners, il nous faut situer leurs pratiques générales, en abordant :

- l'expertise géométrique
- le développement d'outils, de codes et de scripts,
- les enjeux du « workflow »,
- l'assistance à la collaboration entre Design Groups et consultants (externes ou internes),
- les sujets d'intervention et compétences des équipes ARD et SMG en général.

Une fois ces pratiques clarifiées, la présente recherche s'attache plus spécialement à la collaboration entre experts du paramétrique (du SMG et ARD) et Design Teams, ainsi qu'aux aides à la modélisation paramétrique développés au sein de l'agence Foster and Partners.

Les compétences du SMG relatives à la géométrie sont importantes et revendiquées. Néanmoins, elles sont principalement au service de la modélisation informatique par la traduction de formes ou d'intentions, ainsi qu'au service de visées constructives. La plupart du temps, les missions du SMG visent à rationaliser des formes en vue de les rendre constructibles (Whitehead 2009b). D'après Xavier de Kestelier, cela nécessite des bases fortes en géométrie classique c'est-à-dire euclidienne (De Kestelier in Freiburger 2007).

La programmation d'outils et de modèles pertinents pour les Design Team est la mission majeure du SMG et de l'ARD. Pour le SMG, il n'y a pas d'applications « parfaites » permettant de remplir toutes les tâches impliquées dans un projet particulier. Le groupe utilise un grand nombre de logiciels commercialisés en les adaptant, ainsi que des logiciels développés par leurs soins. (Gallou in Kocaturk & Medjdoub 2011, p.242). Mais la compétence la plus importante est d'identifier les outils pertinents à chaque état d'avancement d'un projet (De Kestelier in Kocaturk & Medjdoub 2011, p.242). L'expertise du SMG est avant tout de pouvoir conseiller les équipes de conception ainsi que d'adapter ces outils aux besoins des projets. Ce travail d'adaptation des outils aux nécessités des projets est primordial et intervient tout au long de l'avancement du projet (Whitehead in Kocaturk & Medjdoub 2011, p.238). L'adaptation de la technologie aux projets passe par plusieurs moyens, à différents niveaux :

- la programmation de fonctionnalités par le développement de nouvelles bibliothèques propres à l'agence ;
- l'écriture d'outils spécifiques au projet, sur la base de logiciels existants (comme des plugins) ou bien des logiciels « from scratch » ;
- l'écriture de programmes, scripts ou codes, spécifiques au projet.

---

<sup>36</sup> Le SMG se tient perpétuellement au courant des nouvelles technologies développées afin de les tester et de les intégrer dans l'agence le cas échéant. De Kestelier donne l'exemple de l'intégration dans l'agence, par le SMG, du prototypage rapide. (De Kestelier in Kocaturk & Medjdoub 2011, p.243). Au delà d'un rôle de veille technologique, les équipes SMG (Whitehead in Menges 2006) comme ARD (F. Aish & Chronis 2012) se placent comme des équipes chargées de la Recherche et Développement dans l'agence.

La question du partage et de la réutilisation de ces codes est abordée dans le paragraphe 3.3.3.1.1. Les logiciels, codes ou scripts développés s'inscrivent dans un processus de projet qui doit être pris en compte. La logique de « workflow » est constitutive de l'organisation et du développement des scripts et des programmes.

#### **3.3.1.1.1 Un enjeu majeur : Gérer le « Workflow »**

Le « workflow », littéralement « flux du travail », désigne les différents outils numériques utilisés pour un travail et l'organisation des transferts de données d'un outil à l'autre. Un workflow, c'est-à-dire l'agencement de plusieurs applications logicielles pour un projet précis, peut être centralisé dans un même logiciel regroupant toutes les fonctions nécessaires (comme le propose Digital Project par exemple). Un workflow peut, par ailleurs, solliciter de nombreux logiciels en fonction des tâches à accomplir (2D, 3D, rendu d'image, évaluation, etc.). Le « workflow » rend compte de la complexité et de la multiplicité des tâches des intervenants et des outils utilisés lors d'un processus de projet. La gestion de ce flux est majeure car les différentes applications doivent être compatibles et l'échange de données doit être conçu pour minimiser la perte d'information ou les effets de bruit.

Si l'utilisation d'un modèle peut être une solution contre la perte d'informations, ce n'est pas celle du SMG qui explore plusieurs variantes développées sur des logiciels différents. Le recours à des applications différentes vise, entre autres, à produire des documents adaptés aux différentes missions effectuées par les collaborateurs d'un projet. Pour Whitehead, l'informatique permet d'intégrer de nombreuses informations en recourant à des applications diverses (H. Whitehead in Kocaturk & Medjdoub 2011, p.242). Whitehead s'oppose à l'utilisation d'un unique logiciel BIM, ne convenant ni à la multiplicité des représentations, des médias et des options développées d'un projet, ni à la multiplicité des collaborateurs et de leurs besoins (Whitehead in Kocaturk & Medjdoub 2011, p.246).

Au début de son activité, le SMG s'occupait de développer des outils adéquats (les « customs tools ») au processus de travail des équipes de conception. Aujourd'hui, le SMG peut anticiper le processus global de l'équipe de conception. Les besoins créés par ce processus (en termes de rendus, formats de fichiers, etc.), les capacités et les demandes de l'équipe de conception sont évalués et pris en compte dans la mise en place d'un workflow d'outils numériques adaptés (Whitehead in Kocaturk & Medjdoub 2011, p.239).

Dans une interview avec le chercheur Achim Menges effectuée en 2006, Whitehead résume le processus d'intervention du SMG (Menges 2006, p.45), ainsi: dans un premier temps, le problème est formulé avec l'équipe de conception puis, le SMG propose un workflow adapté aux impératifs de l'équipe de conception (deadline, nature des documents demandés, consultants impliqués sur le projet...). A partir de la définition du workflow, des techniques sont testées et mises en place. La programmation d'outils, scripts ou modèles paramétriques sur mesure y participent éventuellement.

Chez Foster and Partners, les équipes de conception utilisent de nombreux logiciels lors du processus de projet. Elles développent de nombreux modèles pour explorer des options ou des variantes du projet. Cette multiplication des formats et des données rend ardue le partage d'informations entre les collaborateurs. La modélisation paramétrique permet, en théorie, de propager des transformations et donc de conserver la cohérence d'un modèle. Mais aujourd'hui les équipes de conception ne travaillent pas sur un modèle unique mais plutôt sur une « fédération » de modèles (Whitehead in Kocaturk &

Medjdoub 2011, p.240). Ces modèles doivent entretenir des liens les uns avec les autres (des partages de données par exemple) et l'organisation de la propagation des données doit être correctement conçue.

Dans le cas du projet Chesa Futura, l'organisation du travail a permis à l'équipe de conception de travailler sur le même modèle numérique mais sur des zones ou des fichiers différents. Chaque membre était responsable d'un set de géométrie paramétrique, correspondant à une zone du projet (le toit, la structure, etc.). La cohérence de l'ensemble était assurée par le partage de certaines données entre les sets (comme des coordonnées de points ou des surfaces de références) (H. Whitehead in Kolarevic 2005, p.96). Les données partagées doivent être par conséquent stables et peu nombreuses (Whitehead in Kocaturk & Medjdoub 2011, p.241).

La prise en compte des transformations du projet au cours de sa conception est cruciale pour le développement de modèles paramétriques. Quand un modèle est construit, il est souvent difficile de le faire évoluer en fonction des changements introduits dans la conception (cf. Chapitre 2). En effet, un modèle comprend un mécanisme de résolution de contraintes spécifiques souvent difficilement adaptable à un contexte ou des enjeux géométriques différents. Face à cela, le SMG met en place deux stratégies : il investit moins d'énergie dans les premières phases de la conception, et développe des « procédures », des petits modèles ou scripts, réutilisables et adaptables (Whitehead 2009b).

La prise en compte des transformations et des évolutions du projet dépend de la phase d'avancement du projet. Dans les premières phases, les processus technologiques sont très ouverts et susceptibles d'être modifiés ou remplacés. Le processus se stabilise avec le projet. Le SMG privilégie le recours à des « procédures » c'est-à-dire des fragments de processus technologiques susceptibles d'être réutilisés même si le projet évolue (Whitehead 2009b)<sup>37</sup>. Ces procédures, de natures diverses, doivent être suffisamment courtes pour pouvoir être réutilisées dans un autre projet. Le recours à de telles procédures est important, en dépend l'efficacité opératoire du SMG pour optimiser ses développements. Ces procédures sont une aide au workflow et peuvent être utilisées pour la modélisation paramétrique. Le choix d'un recours à la modélisation paramétrique ou le passage d'un modèle paramétrique à un logiciel d'évaluation par exemple, procède des connaissances préalables des procédures.

### ***3.3.1.1.2 Assister la collaboration entre consultants et équipe de conception***

Pour le SMG, les enjeux de la propagation des transformations sont liés à la collaboration entre les différents acteurs de la maîtrise d'œuvre : équipes de conception et consultants, internes comme externes. Les enjeux de partage des tâches et d'échange d'informations influencent la construction des modèles numériques (Whitehead in Kocaturk & Medjdoub 2011, p.236). La construction d'un modèle dépend de ce que l'on en attend (explorer des solutions formelles, générer des objets géométriques spécifiques à une évaluation mécanique, produire un volume productible par une imprimante 3D, etc.).

Outre l'influence technique que représente la nécessité de communiquer les données adéquates aux différents acteurs, la collaboration avec différents experts fait partie de la culture professionnelle du SMG. Ces experts peuvent être internes à l'agence, ce sont le plus souvent des experts en développement durable ou en urbanisme ou, ils peuvent être des consultants externes. Le SMG est

---

<sup>37</sup> « we try to use procedures which are quite small and modular. And then they can be recombining in a way that accelerates the process. » (Whitehead 2009b).

chargé de la communication avec ces spécialistes. Une relation tripartite s'établit alors, entre le SMG, l'équipe de conception et les consultants (Whitehead 2009b).

### **3.3.1.1.3 Missions et compétences des équipes Specialist Modelling Group et ARD**

Sur quels projets interviennent les équipes ARD et SMG? Quels sont les compétences et les formations des membres de ces équipes ? ces questions sont soulevées dans ce paragraphe.

Le SMG intervient dans des cas très variés : les natures des projets comme les phases sont multiples. La longueur des missions et leurs implications dans le projet sont très variables (X. De Kestelier in Kocaturk & Medjdoub 2011, p.237). Sur un même projet, par exemple, l'équipe peut intervenir au début de la conception, puis se retirer du projet pour un temps et revenir ensuite pour intervenir lors de phases de définition plus avancées.

L'équipe du SMG étant très réduite, elle ne peut intervenir sur tous les projets. Le plus souvent l'équipe intervient sur les cas les plus complexes (Whitehead 2009a). Ces projets complexes et «exigeants» peuvent être soumis au SMG par les Partenaires dans le cas de problèmes géométriques ou techniques que l'équipe de conception n'arrive pas à résoudre seule. Cependant, dans ces cas là, pour le SMG, c'est souvent «trop tard» (Whitehead 2009b). Le SMG préfère intervenir dès les premières phases de la conception (De Kestelier in Kocaturk & Medjdoub 2011, p.237) pour accompagner les choix : prévenir les impossibilités géométriques ou techniques, ou bien encore «*ouvrir de nouveaux territoires*» (Whitehead 2009b). C'est pourquoi le SMG n'intervient pas seulement «à la demande» comme support quand un problème se présente, mais il intervient également de lui-même sur un projet, s'il estime cela pertinent. Cette intervention peut être proposée par un Partenaire (F. Aish & Chronis 2012) ou par un membre du SMG lors du «Design Boards» (Whitehead 2009b). Le «Design Boards» est une revue d'avancement des projets.

Dans le cas du projet de couverture de la cour du Smithsonian Institute (en 2004) par exemple, le SMG est intervenu dès les premières phases de conception pour conseiller les équipes sur les enjeux (Menges 2006, p.48). Aujourd'hui, les membres des équipes de conception ainsi que les Partenaires sont sensibilisés aux outils numériques et il leur arrive souvent de solliciter le SMG ou l'ARD (F. Aish & Chronis 2012).

Pour remplir la diversité de ces missions, les membres du SMG et de l'ARD ont des profils particuliers. Les compétences des membres du SMG et de l'ARD sont très variées. Tous les membres, architectes ou ingénieurs de formation, possèdent une solide expérience en architecture. C'est cette expérience qui est la plus valorisée, en outre de rassembler des compétences multidisciplinaires (spécialistes de structures gonflables, de l'ingénierie aéronautique, ou encore artistes) (Whitehead in Kolarevic 2005, p.83; F. Aish & Chronis 2012). Whitehead qualifie ses collègues d'«Architects plus» (Whitehead 2009a; Kocaturk & Medjdoub 2011) car tous sont architectes et ont développé des compétences spécifiques à la programmation, la géométrie, etc. Les membres du SMG ont de solides connaissances en programmation informatique et manipulent tous plusieurs langages de programmation à différents niveaux (Whitehead 2009b). La formation nécessaire aux membres du SMG (ainsi que du SMGE et ARD par extension) est très exigeante. Pour Whitehead il faut deux ans de pratique pour être réellement opérationnel au sein du SMG, un an d'expérience pour assimiler les compétences et les connaissances techniques et un an pour pouvoir interpréter correctement les intentions des équipes de conception (Whitehead in Kocaturk & Medjdoub 2011, p.239). C'est finalement la capacité à interpréter

les pensées et les intentions de l'équipe de conception qui semble être la compétence la plus exigeante et la plus demandée.

### **3.3.2 Collaborations et hybridations entre le SMG et les Design Teams**

La collaboration entre les équipes de conception et les experts des SMG et ARD est majeure. Elle pointe les porosités et les « hybridations » (cf. 3.2.3) possibles entre modélisation paramétrique et conception architecturale.

Le SMG doit faire exprimer aux équipes de conception leurs intentions et leurs besoins (Whitehead in Kocaturk & Medjdoub 2011, p.238). Une relation privilégiée entre le SMG et les équipes de conception est donc nécessaire. Elle permet de comprendre les intentions du projet, d'intégrer les contraintes (de rendus, de formats de fichiers, *etc.*) et d'utiliser au mieux les compétences techniques de l'équipe de conception. Tout cela sert à l'intégration pertinente de la modélisation paramétrique dans le développement du projet.

Cette relation passe souvent par la collaboration. Mais elle peut également procéder de l'hybridation de compétences entre les membres des équipes de conception et les membres du SMG.

Des tentatives d'intégration des membres du SMG aux équipes de conception ont été effectuées mais n'ont pas toujours abouties. Aujourd'hui, cette stratégie n'est plus entretenue, elle n'est pas nécessaire : les équipes de conception sont suffisamment familiarisées avec les outils du SMG et de l'ARD (F. Aish & Chronis 2012) et ont une connaissance de leurs potentialités qui leur permet de les solliciter en cas de besoin.

### **3.3.3 Assistances à la modélisation paramétrique développées chez Foster and Partners**

L'utilisation de la modélisation paramétrique pour la conception architecturale n'est pas toujours facile, même pour un expert. Chez Foster and Partners, des aides à la modélisation paramétrique sont développées par le SMG et l'ARD, en interne ou pour les membres des équipes de conception.

#### ***3.3.3.1.1 Au sein des SMG et ARD pour leur propres pratiques : développer des aides techniques et une culture***

Le SMG et le groupe ARD développent, depuis quelques années, divers outils pour assister leurs pratiques, qui sont constitués principalement de programmes (outils, codes ou scripts) réutilisables. Dès 2009, une « banque » de ressources informatiques (codes, plugins, bibliothèques) est développée par Francis Aish. Cette « banque » structure les ressources, les rend accessibles et promeut certaines idées ou méthodes de programmation (Whitehead 2009b).

Cette banque permet:

- de gagner du temps en réutilisant des éléments existants (F. Aish & Chronis 2012),
- d'apprendre à modéliser en ayant accès à des ressources (Whitehead in Kocaturk & Medjdoub 2011, p.239). Les modalités d'apprentissage par l'utilisation de modèles ou de codes existants sont interrogées dans la suite de cette thèse.

La réutilisation de codes ou de modèles paramétriques pose divers problèmes. Un code doit être réorganisé, voire réécrit, pour être lisible par un utilisateur extérieur (Woodbury 2010). Par ailleurs, le partage de codes demande de penser les modalités d'accès et les droits d'écriture de chacun : qui peut modifier le code ? Qui peut voir le code source ? Le parti pris de Francis Aish pour l'ARD est de partager en interne tous les codes, mais de ne partager avec les membres des Design Teams que les Applications, les bibliothèques et les API. (F. Aish & Chronis 2012). L'ARD garde ainsi le « contrôle » sur les outils diffusés (F. Aish & Chronis 2012). Outre le partage de technologies, des stratégies de partage de savoirs sont mises en œuvre par l'organisation de « sessions » (Whitehead 2009b). Ces stratégies de partage des connaissances et de ressources pour l'aide à la modélisation paramétrique nous intéressent tout particulièrement dans cette thèse (cf. chapitre 4).

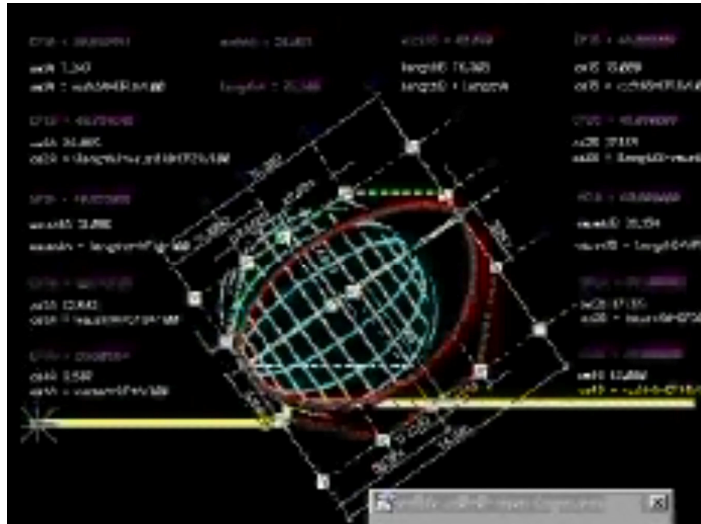
### ***3.3.3.1.2 Assistance à l'usage de la modélisation paramétrique par les équipes de conception***

Dans la pratique de Foster and Partners, différents moyens sont mis en œuvre pour rendre les programmes et modèles développés accessibles au plus grand nombre (collaborateur, maître d'ouvrage, équipes de conception, partenaires, entreprise, *etc.*). Ces moyens sont : la production de documents d'aide (dossiers, guides, help files, *etc.*), le développement de programmes sur des supports connus de l'utilisateur et, le travail des interfaces.

Le plus souvent possible, les membres d'ARD développent des programmes basés sur les plateformes habituelles des équipes de conception que sont : MicroStation, Generative Components, Rhinocéros. L'enjeu est d'augmenter l'efficacité du travail de tous les membres de l'agence, jusqu'aux collaborateurs les moins experts en technologies numériques (F. Aish & Chronis 2012).

Le développement d'interfaces faciles d'utilisation est un axe fort du travail d'ARD (F. Aish & Chronis 2012). De la même façon, le SMG s'attache à produire des interfaces et des paramètres manipulables simplement, voire intuitivement (comme avec des « sliders » où l'on transforme une valeur en déplaçant un curseur sur une graduation). Les modèles paramétriques sont ainsi construits pour que les paramètres (variables indépendantes) soient peu nombreux et facilement lisibles. De même, les résultats produits par un modèle paramétrique (géométries, listes de valeurs issues d'une évaluation) sont contraints pour plus de lisibilité (représentation graphique, peu de textes) (F. Aish & Chronis 2012). Dans le cas du modèle proposé pour la conception du City Hall par exemple, Whitehead décrit la conception du modèle par une recherche de paramètres rendant le modèle paramétrique facilement manipulable : pour la définition de la géométrie, seuls quelques points que l'on peut faire glisser dirigent la courbure de la surface (cf. Figure 40) (Whitehead 2009).





**Figure 40 : Modèle paramétrique développé pour la conception du City Hall [fosterandpartners]**

Ces interfaces, qualifiées par les acteurs du SMG et de l'ARD d'« intuitives », peuvent également viser la communication d'un modèle à des acteurs extérieurs (entreprises, maître d'ouvrage, etc.) (Kocaturk & Medjdoub 2011, p.232). Dans ces cas là, les interfaces peuvent prendre la forme de Touch Screen, de projections visant à « augmenter » des maquettes physiques, *etc.* (F. Aish & Joyce 2012).

Un des enjeux du SMG en 2009 était également de former les équipes de conception en vue de mieux les assister (Whitehead in Kocaturk & Medjdoub 2011, p.239). En 2012, les architectes collaborateurs de chez Foster and Partners sont de plus en plus formés à ces outils.

### **3.3.4 Enjeux de la modélisation paramétrique pour le SMG**

Le SMG est garant d'une expertise technologique permettant de maîtriser des géométries dites complexes, d'explorer rapidement des solutions et de communiquer facilement avec consultants et entrepreneurs. Pour Whitehead, la multiplication des représentations du projet dès les premières phases du projet grâce à l'intégration des nouvelles technologies, facilite la communication avec les partenaires lors des « Design Board » ainsi que la prise de décisions (Whitehead in Kocaturk & Medjdoub 2011, p.243). De nombreux points de vue sur le projet peuvent être présentés et explorés *via* la diversité des représentations (perspectives, évaluation des performances, maquettes physiques augmentées, etc.) et, des modifications géométriques peuvent être explorées en temps réel lors du « Design Board ».

Le SMG valorise la créativité en n'imposant pas une technologie mais en définissant une stratégie technologique en fonction du projet impliqué (Gallou in Kocaturk & Medjdoub 2011, p.238), en proposant, entre autres, des technologies nouvelles. De Kestelier donne l'exemple de l'introduction de l'impression 3D au sein de l'agence, aujourd'hui très utilisée. (De Kestelier in Kocaturk & Medjdoub 2011, p.243). L'expertise technologique du SMG est non seulement introduit dans la conception de nouveaux outils (programmes, plugins, pour la représentation et l'évaluation, mais aussi pour la génération et l'optimisation), mais vise aussi à « ouvrir de nouveaux territoires » (Whitehead 2009b) et « de nouveaux paramètres » (Gallou in Kocaturk & Medjdoub 2011, p.238).

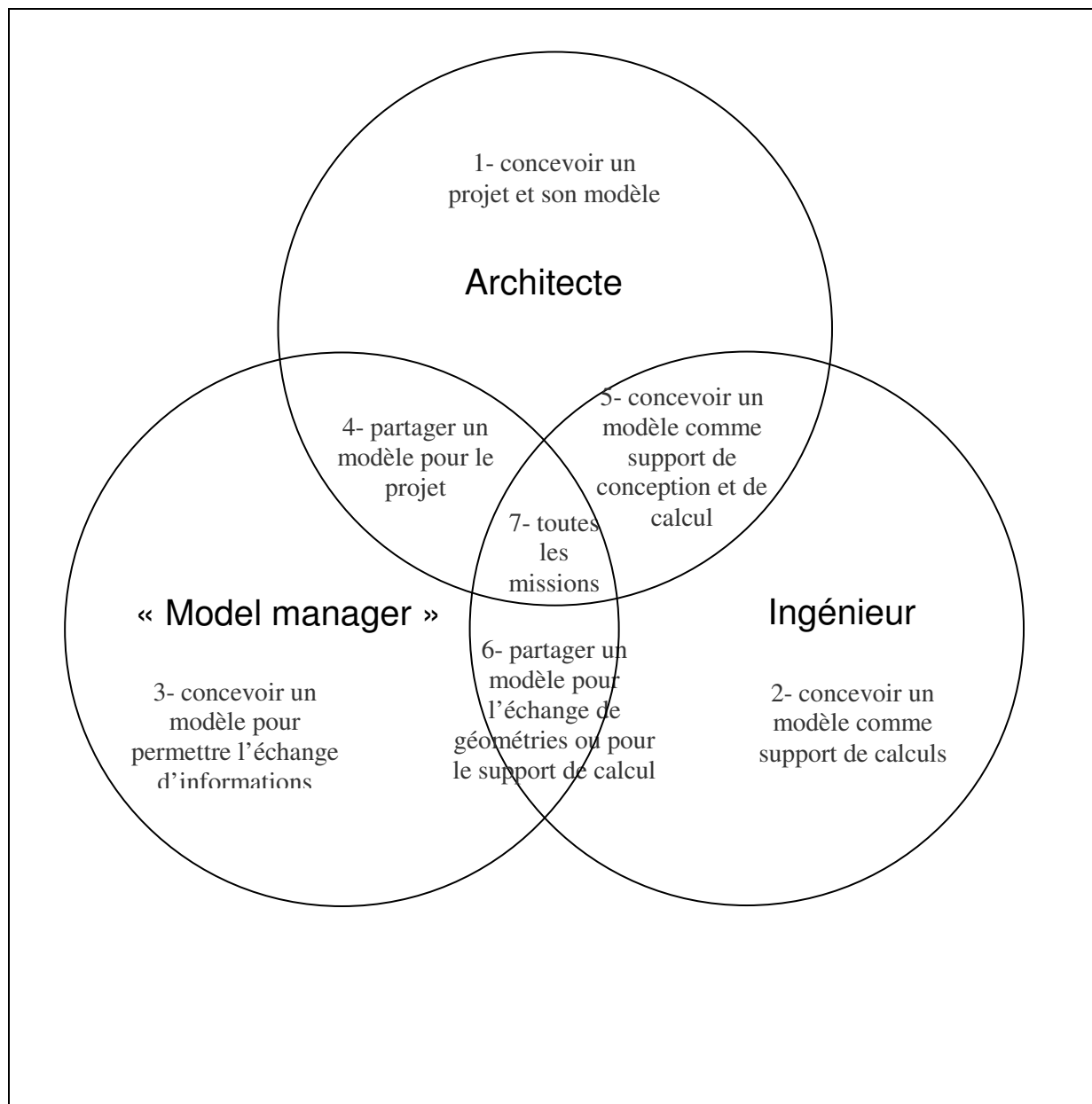
### **3.4 La profession d'architecte : la modélisation paramétrique comme nouveau métier ou comme extension de l'expertise de l'architecte ?**

Nos analyses ont permis d'observer des missions spécifiques à la modélisation paramétrique ainsi qu'une évolution des pratiques des architectes. Ces missions spécifiques amènent à s'interroger sur le travail de l'architecte. Si cette interrogation n'est pas l'objet de notre thèse, nous proposons ici de la soulever succinctement en vue de situer l'apport de l'apprentissage de la modélisation paramétrique par les architectes.

Dans son ouvrage *« Etre Architecte : Les Vertus de l'Indétermination. De la sociologie d'une profession à la sociologie du travail professionnel »*, Olivier Chadoin interroge le statut d'architecte (2006). Pour Chadoin, la profession d'architecte se transforme par réduction de l'expertise de l'architecte (ce qu'il nomme redéfinition par « rétrécissement » (Chadoin 2006, p.24)) et, la profession se transforme aussi par extension des missions de l'architecture. Cette dernière redéfinition implique une extension de l'expertise mais toujours sur la base d'une formation généraliste de l'architecte (Chadoin 2006, p.25). Pour l'auteur, le contexte actuel pousse à l'éclatement d'une mission globale de l'architecte en missions partielles, dans lequel on observe à la fois : la création de nouveaux postes liés à la profession d'architectes et l'association de la profession d'architecte avec de nouveaux partenaires. Nous l'avons vu précédemment, les pratiques de la modélisation paramétrique sont diverses (représentation de géométries complexes, aide à la fabrication, à l'évaluation ou à l'optimisation, etc.) et, on peut s'interroger sur le positionnement de ces pratiques comme « extension des compétences » de l'architecte ou comme création de nouveaux métiers partenaires.

#### **3.4.1 Les pratiques liées à la modélisation paramétrique**

Pour avancer dans la compréhension du rôle des pratiques de la modélisation paramétrique dans le paysage de la profession d'architecte, on peut situer les différentes pratiques abordées lors de cette analyse par rapport aux trois principales polarités qui semblent s'y rapporter : celles des professions d'architecte, d'ingénieur et de « model manager » (Smith 2005; Sheldon 2002; Eastman et al. 2008; Bonora 2012). Nous proposons d'interroger les interconnexions et les recouvrements de ces pratiques au travers de la représentation de ces polarités dans un diagramme de Venn (cf. Figure 41).



**Figure 41 : Croisement des pratiques de modélisation paramétrique autour des professions d'architecte, d'ingénieur et de model manager**

Dans la Figure 41, nous installons les pratiques de la modélisation paramétrique à l'interface des compétences de l'architecte, de l'ingénieur et du *model manager*. Grâce à la formalisation de ces polarités en un diagramme de Venn, nous pouvons identifier les relations entre ces trois ensembles de compétences. Ces relations permettent de situer sept typologies de pratiques :

- 1- Concevoir un modèle paramétrique en lien avec une activité de conception architecturale :  
Cette situation, observée précédemment, est encore rare.
- 2- Concevoir un modèle paramétrique comme support de calcul :  
C'est par exemple les pratiques menées par les membres de RFR et TESS pour le projet du Pavillon Louis Vuitton (cf. 3.2.2).

- 3- Concevoir un modèle paramétrique pour permettre l'échange d'informations :

Peu étudiées dans le cadre de cette thèse, ces pratiques sont observées par exemple dans le cadre d'une des missions de Decode avec le CSTB pour le projet Canopée des Halles (Bonora 2012) ou d'une des missions de Gehry Technologies pour le projet du Pavillon pour la Fondation Louis Vuitton pour la Création (Witt 2008).

- 4- Concevoir un modèle paramétrique pour assister la collaboration dans le cadre de la conception architecturale :

Nous n'avons pas abordé cette pratique, mais nous pouvons citer comme exemple le travail de l'agence Decode (Bonora 2012).

- 5- Concevoir un modèle paramétrique comme support partagé de conception et de calcul :

Cela a par exemple été le cas de la mission du Specialist Modelling Group lors du projet du City Hall, où un modèle a été développé pour que l'équipe puisse définir une instance de forme adaptée à leurs intentions puis la faire évaluer par un BET du point de vue de l'ensoleillement (Whitehead 2009b).

- 6- Partager un modèle paramétrique pour l'échange de géométries ou pour le support de calcul :

Lors de ces pratiques, le modèle vient représenter un projet déjà conçu, comme c'est le cas de l'intervention de Decode sur le projet de Canopée les Halles à la demande du BET structure où un modèle paramétrique Grasshopper a été développé en vue de contrôler les géométries et en calculer les performances.

- 7- Concevoir un modèle paramétrique pour à la fois supporter la conception architecturale, les calculs de performance mais aussi pour assister le partage d'informations :

Cette pratique semble être menée par le Specialist Modelling Group pour certains projets. Pour Guy Tapie, la spécialisation des acteurs ne s'oppose pas à l'émergence de certains profils généralistes, capables de gérer à la fois la complexité des logiciels comme celle de la conception architecturale ou de certaines exigences techniques (Tapie 2000, p.96). Ces rares profils résultent d'une longue expérience de l'individu au contact de ces différentes expertises (ce qui s'observe dans les parcours de Dennis Shelden, Francis Aish ou Hugh Whitehead par exemple).

Il faut noter ici que chacune des pratiques peut être menée par des structures différentes (agence autonome, équipe de consultants internes, pratique spontanée d'un individu au sein d'une agence, etc.). On peut donc penser que chacune de ces missions peut donner lieu à la création d'une nouvelle profession mais, également que ces missions peuvent-être remplies par un « architecte », jusqu'à la possibilité, rare mais existante, de rencontrer un « architecte-ingénieur-model manager ». Mais cela est-il souhaitable pour la profession d'architecte ?

### 3.4.2 Un architecte aujourd'hui

Définir ce qu'est, pourrait ou devrait être un architecte aujourd'hui, est une tâche presque impossible, pourtant, soulever cette question est incontournable lorsque l'on souhaite proposer une pédagogie en vue de former de futurs architectes.

Pour Olivier Chadouin, au-delà des nouvelles compétences et nouvelles missions, la profession d'architecte persiste : celle-ci étant hétérogène. Il reprend la thèse de Tapie, affirmant que « *l'hétérogénéité des savoirs et des compétences* » et « *l'hybridation des compétences* » est constitutive de la profession d'architecte (Tapie 2000). Pour Chadouin, la profession d'architecte se caractérise par une « indétermination », permettant à l'architecte de se renégocier et réinventer sa position et sa pratique (Chadouin 2006, p.33). Intégrer des compétences nouvelles (l'ingénierie pour les ingénieurs-architectes ou la programmation pour les « architectes plus » (Whitehead 2009b)) n'est donc pas une dissolution de l'architecte, mais s'inscrit pleinement dans l'*indétermination* de celui-ci.

Par ailleurs, nous l'avons vu dans la précédente analyse des pratiques, les intérêts et les besoins auxquels répond la modélisation paramétrique sont nombreux. Ces besoins sont : la représentation et le contrôle de géométries complexes, la rapidité d'exécution de représentation, la rapidité d'exploration de variantes d'une forme, la rapidité d'évaluation ou encore, l'aide à la fabrication ou à la coopération grâce aux modeleurs paramétrique par objets (Eastman et al. 2008). Ces besoins sont cruciaux et si l'architecte veut rester compétitif, il a un réel intérêt à pouvoir y répondre. Etre capable de concevoir ses propres modèles paramétriques représente, selon nous, un atout pour l'architecte. C'est la raison pour laquelle l'apprentissage de la modélisation paramétrique semble avoir toute sa place au sein des écoles d'architecture.

Si l'apprentissage de la modélisation paramétrique est un atout pour les futurs architectes (cf. 3.2 et 3.3), nous avons par ailleurs souligné les difficultés d'usages que posent les modeleurs paramétriques (cf. 2.3) pour ces étudiants et pour les praticiens qui les utilisent. C'est pourquoi dans le Chapitre suivant, nous proposons d'interroger les aides existantes à la modélisation paramétrique, destinées à faciliter la conception de modèles paramétriques par les architectes.



## 4 Les aides à la modélisation paramétriques existantes

Les modeleurs paramétriques demandent des compétences spécifiques pour concevoir la représentation *symbolique* d'une géométrie (*cf.* Chapitre 2). Ceci représente une difficulté pour des non-spécialistes. Pourtant, les usages actuels des modeleurs paramétriques deviennent de plus en plus importants (Kolarevic 2005; Szalapaj 2000; Picon 2010). Ce contexte d'usages grandissant de la modélisation paramétrique, nous amène à nous intéresser aux dispositifs existants d'aide à l'usage des modeleurs paramétriques.

Plusieurs recherches s'intéressent à l'usage de la modélisation paramétrique par les architectes (Davis et al. 2011; Harding et al. 2012; Chien & Yeh 2012). Mais elles n'ont pour la plupart pas produites elles-mêmes d'aides complémentaires. Le travail de Robert Woodbury et de son équipe, quant à lui, semble donner lieu à une aide effective : la méthode des Patterns, que nous explicitons en 4.1.

De nombreux autres acteurs de l'architecture (étudiants, architectes, ingénieurs, experts de la modélisation paramétrique, etc.) s'intéressent à l'aide à la modélisation paramétrique et partagent leurs productions sur internet. Nous nous intéressons à cette production qui peut constituer une aide à la modélisation paramétrique dans le paragraphe 4.2 de ce Chapitre.

### Plan du chapitre :

|            |  |            |
|------------|--|------------|
| <b>4.1</b> | <b>Une aide à la modélisation paramétrique en architecture : La méthode des Patterns</b> | <b>106</b> |
| 4.1.1      | Qu'est ce qu'un « Pattern » ? de l'architecture à l'informatique                         | 106        |
| 4.1.2      | Patterns et modélisation paramétrique : la proposition de Robert Woodbury                | 113        |
| <b>4.2</b> | <b>Supports en ligne : informations en libre accès.....</b>                              | <b>122</b> |
| 4.2.1      | Les supports en ligne  | 122        |
| 4.2.2      | « <i>Catalogues</i> » de <i>samples en ligne</i>   | 129        |
| 4.2.3      | Présences en ligne et effets de communautés  | 138        |

## **4.1 Une aide à la modélisation paramétrique en architecture : La méthode des Patterns**

Un des axes de recherche du laboratoire « Computational Design Canada », dirigé par Robert Woodbury, propose le développement d'une aide à la modélisation paramétrique destinée aux architectes : la méthode des Patterns (Marques & Woodbury 2006; Marques & Woodbury 2007; Qian et al. 2007; Woodbury 2010). Le concept de « Pattern », construit par Christopher Alexander pour l'architecture (Alexander 1977; Alexander 1999), a été repris par d'autres disciplines telles que l'informatique pour développer des méthodes d'assistances à la conception de programmes orientés-objet (Gamma et al. 1993) ou ici à la conception de modèles paramétriques par les architectes. Dans ce présent Chapitre, le paragraphe 4.1.1 situe la genèse de ce concept en architecture et en programmation et, le paragraphe 4.1.2 explicite plus spécifiquement son appropriation par Woodbury comme outil pour l'aide à la conception de modèles paramétriques.

### **4.1.1 Qu'est ce qu'un « Pattern » ? de l'architecture à l'informatique**

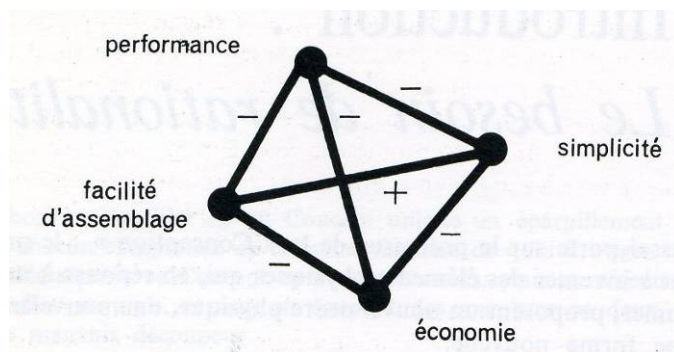
« Pattern » peut signifier en français « *modèle, exemple, échantillon, type, patron (en couture), motif* » en fonction du contexte dans lequel il est employé (Arnold 1977, p.154). La notion de « Pattern », utilisé par Woodbury, s'appuie sur celle développée par Christopher Alexander dans les années 70. Les Patterns sont proposés par Christopher Alexander pour désigner des stratégies visant à résoudre des problèmes précis dans le champ de la conception architecturale (Alexander 1971). Ce sont des outils méthodologiques qui permettent de comprendre pour apprendre à faire de l'architecture. Selon Woodbury, les Patterns existe en architecture, depuis bien avant le travail d'Alexander, dans les traités comme *I Quattro Libri dell'Architettura* d'Andrea Palladio (Woodbury et al. 2007). Après son développement en architecture par Alexander (Alexander 1971; Alexander 1966; Alexander 1977; Alexander 1999), la notion de Pattern a été reprise par d'autres disciplines dont, en particulier, l'informatique (Gamma et al. 1993). La notion de Pattern, dans l'acception retenue ici, est aussi appelée « design Pattern ». L'organisation de Patterns en un tout lié à l'expertise d'une discipline constitue un « Pattern Language » (Alexander 1977).

#### **4.1.1.1 La notion de Pattern chez Christopher Alexander**

Nous nous attachons ici à comprendre et exposer la notion de Pattern, fondatrice, construite par Christopher Alexander et son équipe. Pour cela, nous nous appuyons sur les ouvrages d'Alexander *The synthesis of form, Essay* (en version française *De la synthèse de la forme, essai*) de 1971 et *A Pattern language, Towns, Building, Construction* de 1977, ainsi que sur l'article plus tardif « The Origins of Pattern Theory: The Future of the Theory, and the Generation of a Living World » de 1999. De plus, nous nous appuyons sur le travail de Madeleine Arnold publié en 1977 dans l'ouvrage *Les modèles chez Alexander, Approche critique du Pattern Language*, celui-ci s'attachant à une étude fine des récurrences et des définitions des termes « Patterns » et « Patterns Languages » dans les écrits d'Alexander.



Pour Alexander la conception de la forme, à laquelle il ramène la conception architecturale, consiste en la recherche d'une adaptation correcte entre une forme et un contexte. Cette adaptation est un équilibre entre le diagramme des forces du contexte et le diagramme, complémentaire, de la forme (Alexander 1971, p.17). A partir de là, Alexander propose de penser un système imaginaire « agissant » dans un processus de fabrication de la forme. Ce système peut être représenté par un graphe dont les variables sont les conditions qui doivent être remplies par la « bonne adaptation » entre la forme et le contexte. Les interactions entre les variables sont les relations causales qui les lient les unes aux autres (Alexander 1971, p.35). Ce graphe peut être représenté avec des points pour les variables et des arcs pour les interactions (*cf.* Figure 42 et Figure 43). Comme exemple de graphe, Alexander propose celui du cas d'une habitation n'ayant pas assez de lumière naturelle. Des fenêtres peuvent être ajoutées, mais mettent en cause l'intimité, voire la stabilité structurelle de l'édifice. Eclairage, intimité et stabilité seraient ici trois points variables du système, reliés par des relations causales. Ou encore, Alexander donne l'exemple de la conception d'un aspirateur et, plus particulièrement, du choix de ses matériaux : performance structurelle et fonctionnelle, coût de production et facilité d'assemblage des pièces sont autant de variables en interaction à pondérer lors de la conception de la forme, comme le formalise la Figure 42 (Alexander 1971, p.2). Pour Alexander, la conception se pose en termes de problèmes, où des exigences, parfois en contradiction, sont à satisfaire (Alexander 1971, p.2). Cela justifie pour lui une modélisation de la conception en termes de variables et d'interactions pondérées et causales entre ces variables.



**Figure 42 : Système de conception graphe de force avec lignes conflictuelles (Alexander 1971, p.2)**

Alexander s'intéresse à la manière d'aider un concepteur à résoudre des problèmes de conception, c'est-à-dire à assister la recherche d'une forme qui tendrait vers une adaptation à son contexte. Cette résolution n'est pas la recherche d'un optimum mais la recherche d'une forme visant la « non-inadaptation » à un contexte (Alexander 1971, p.34). Malgré la multiplication des variables et de leurs interactions (*cf.* Figure 43), pour Alexander, des sous-systèmes distincts quasiment indépendants (*cf.* Figure 44), peuvent être identifiés. Ces sous-systèmes sont dus aux différences d'intensité des interactions causales entre les variables (*cf.* Figure 42). Cette représentation permet de visualiser le « domaine » des inadaptations de la situation de conception (Alexander 1971, p.67).

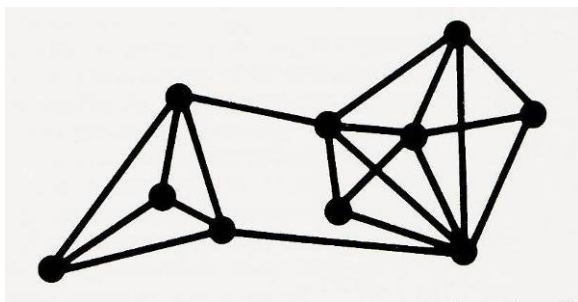


Figure 43 : Exemple de « système « agissant » dans un processus de fabrication de la forme » (Alexander 1971, p.35)

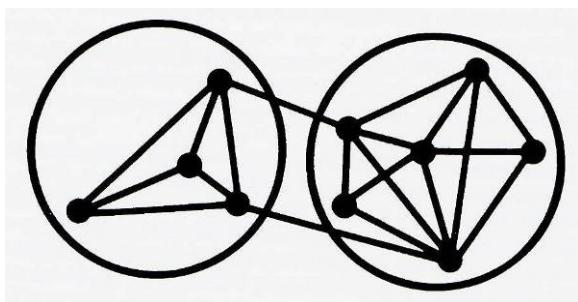


Figure 44 : Exemple de décomposition d'un système en sous-systèmes indépendants (Alexander 1971, p.36)

Cette distinction en sous-systèmes (cf. Figure 44) réduit la complexité du système global et le rend solvable pas à pas. Cette décomposition est, pour Alexander, la seule possibilité de résolution de problèmes complexes dans des délais raisonnables (Alexander 1971, p.34).

Alexander propose une méthodologie de conception visant à accompagner l'architecte dans la formalisation d'un problème et la résolution de celui-ci *via* une forme non-inadaptée. Cette méthode fait appel à une phase d'analyse du contexte et de ces exigences. Ces exigences sont décomposées de manière à construire un « programme » utile pour la conception (Alexander 1971, p.99). Ce programme est synthétisé sous la forme d'un diagramme (Alexander 1971, p.71), où les exigences sont les « variables d'inadaptations possibles » (Alexander 1971, p.50). Ces variables peuvent être regroupées (cf. Figure 45) et identifiées en « concepts », comme « économique », « acoustique », « la sécurité » (Alexander 1971, p.54) : ces regroupements permettent au concepteur d'appréhender plus facilement l'ensemble de la situation de conception (cf. Figure 45).

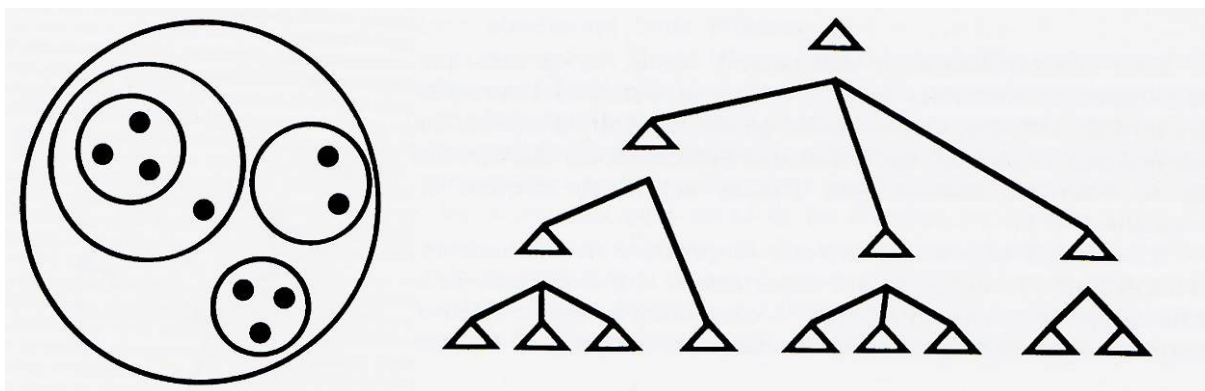


Figure 45 : Décompositions de variables : deux exemples de formalisation (Alexander 1971, p.68)

Pour Alexander, ce nouveau découpage du système par groupes de variables ne correspond pas forcément au découpage du système en sous-systèmes indépendants, tels que nous l'avons défini précédemment. Ainsi, on observe deux découpages différents entre la Figure 44, exemplifiant un découpage en sous-systèmes indépendants, et la Figure 46, exemplifiant un regroupement de variables liées à un même « concept ».

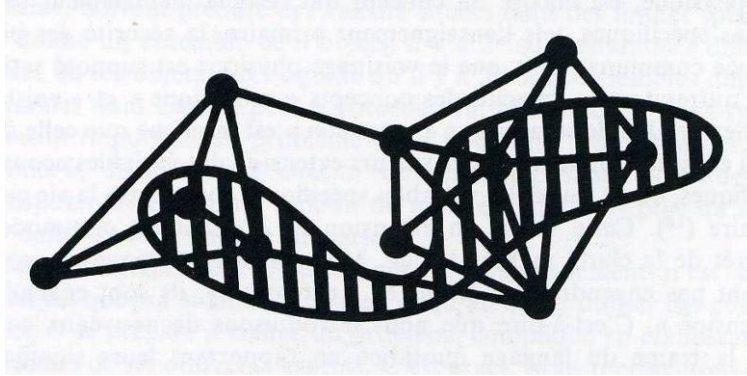


Figure 46 : Nouveau découpage proposé: regroupement de variables par "concepts" (Alexander 1971, p.55)

Du diagramme<sup>38</sup> analytique des exigences, découle le diagramme synthétique de sa réalisation dans une forme (Alexander 1971, p.78). Chaque variable peut être adaptée ou non. Le diagramme de programme permet de définir un domaine D de formes, pour lesquelles les variables peuvent être évaluées comme étant adaptées (= 1) ou inadaptées (= 0). Les formes définies par D sont alors évaluées en fonction de leur non-inadaptation (Alexander 1971, p.86). Dans la Figure 47 par exemple, le domaine D, défini à partir de deux variables  $X_i$  et  $X_j$ , est divisé en quatre ensembles dont un seul contient des formes adaptées.

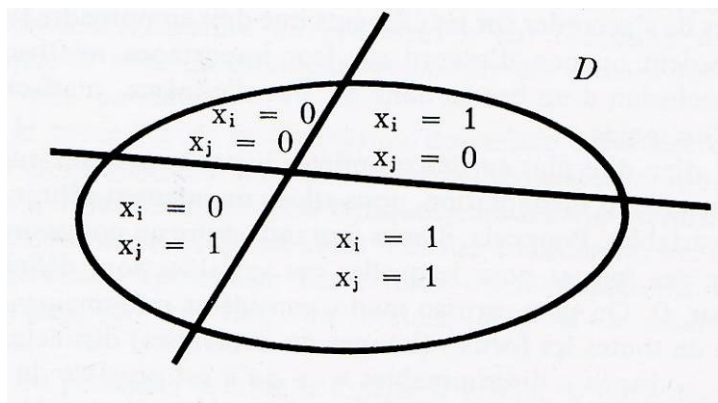



Figure 47 : Domaine de formes D et évaluations de l'adaptation ou non-adaptation de ses variables (Alexander 1971, p.88)

Les sous-systèmes de variables de non-inadaptation sont identifiables au sein d'un graphe de conception (cf. Figure 44). Pour Alexander, certains problèmes ou conflits de forces sont génériques et récurrents (cf. Figure 46). C'est pourquoi, il propose de construire des typologies de solutions, appelées

<sup>38</sup> Les caractéristiques et les rôles précis de ces diagrammes de « programme » sont explicités dans *De la synthèse de la forme, essai* dans le Chapitre « Solution » que nous n'aborderons pas dans le détail ici (Alexander 1971, p.99-110).

*Patterns* (Arnold 1977, p.15). Un Pattern est une relation spécifique entre des variables d'inadaptation empêchant un conflit déterminé (Arnold 1977, p.17). Un Pattern n'est pas un optimum, mais une solution de non-inadaptation (Arnold 1977, p.6). Dans la méthodologie d'Alexander, diagrammes de problèmes et diagrammes de solutions sont liés (Alexander 1971, p.78). Chaque sous-système « Pattern » rassemble donc à la fois un problème, un contexte et une solution (Arnold 1977, p.12). Ainsi chaque Pattern est structuré : - d'une description du problème, - d'une description de la solution proposée et de sa pertinence, ainsi que - du contexte dans le lequel le Pattern est utilisable.

Un *Pattern language*, quand à lui, est une méthodologie visant à aider l'élaboration du *programme*, c'est-à-dire du système de variables d'inadaptation (Arnold 1977, p.34). Un *Pattern Language* rassemble des Patterns appartenant à un même champ d'expertise. Alexander et son équipe mettent en application leur méthode de conception en proposant de nombreux Patterns constituant un *Pattern Language* spécifique à la conception de l'architecture et de la ville. Ce *Pattern Language* est publié dans l'ouvrage *A Pattern language, Towns, building, construction*<sup>39</sup> (1977) dans lequel sont rassemblés deux cents cinquante trois Patterns. Ces Patterns portent sur des problèmes génériques et peuvent contenir des prescriptions, comme le montre par exemple le Pattern n°61, décrit dans le Tableau 7 (Alexander 1977, p.311). Dans ce pattern concernant le dimensionnement d'un espace public, une prescription est donnée : une place ne doit pas dépasser entre 45 et 60 pieds de large.

|   |  |
|---|--|
| <p>Pattern n°61 :<br/>« Small Public Square »</p> | <p>Problem: « <i>a town needs public square; they are the largest, most public rooms, that the town has. But when they are too large, they look and feel deserted.</i> » (Alexander 1977 p.311)</p>  |
|   | <p>Solution: « <i>Make a public square much smaller than you would at first imagine; usually no more than 45 to 60 feet across, never more than 70 feet across. This applies only to its width in the short direction. In the long direction it can certainly be longer</i> » (Alexander 1977 p.313)</p> |
|   |  <p style="text-align: right;">(Alexander 1977 p.313)</p>   |

**Tableau 7 : Pattern 61 "Small public squares" d'après (Alexander 1977, p.311)**

<sup>39</sup> Il faut noter que cet ouvrage est présenté comme indissociable de *The Timeless Way*, ce dernier présente la discipline architecturale et urbaine telle que formalisée par Alexander et son équipe, ainsi que la manière d'utiliser les Patterns décrit dans *A Pattern language, Towns, building, construction*.

Tous les Patterns sont reliés entre eux : ils se relancent les uns les autres et forment le « Pattern Language ». Les interrelations entre Patterns sont très importantes car aucun Pattern ne peut exister isolément (Alexander 1977, p.xiii). Dans le système d'Alexander, comme le montre la Figure 48, le Pattern Language est mis en jeu à deux niveaux: celui de sa production et celui de son utilisation (Arnold 1977, p.34). Toute personne peut proposer un nouveau Pattern ou proposer une critique d'un Pattern existant (Arnold 1977, p.35). Pour Alexander, les Patterns sont à un niveau d'abstraction suffisant pour permettre le passage de solutions existantes, ayant permis la définition des Patterns, à des solutions futures ouvertes et potentiellement toujours différentes en fonction des contextes (cf. Figure 48).

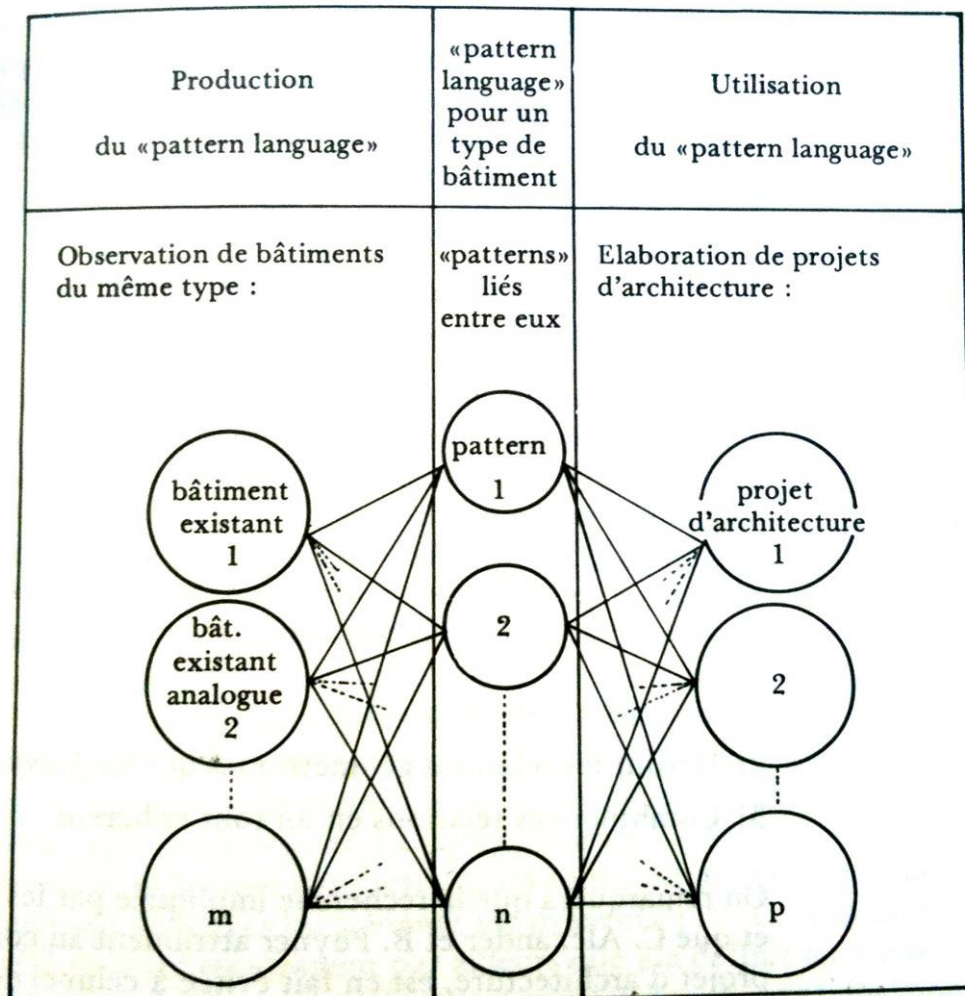


Figure 48 : Production et utilisation d'un "Pattern Language", (Arnold 1977, p.35)

Finalement, chaque Pattern est une solution tactique apportée à un problème circonscrit à un contexte et récurrent dans un domaine d'expertise. La relation d'un Pattern à la conception pose question puisqu'elle suppose de considérer la conception comme une activité de résolution de problèmes, ce qui n'est pas le paradigme que nous suivons dans cette thèse (cf. Chapitre 5)<sup>40</sup>. Cependant les Patterns ont

<sup>40</sup> Nous nous appuyons dans la présente recherche sur les apports de l'architecturologie pour étudier la conception comme opérations d'attribution de mesures (Boudon 1971; Boudon 2004; Boudon et al. 2000).



un rapport intéressant à la stratégie : un Pattern doit être suffisamment général pour ne pas induire la conception tout en apportant une réelle aide. Ainsi, pour Alexander, un même pattern peut être utilisé de façon différente un nombre infini de fois sans jamais produire la même solution (Alexander 1977)<sup>41</sup>. Cependant, nous l'avons vu, avec le Pattern n°61, ces solutions semblent être relativement prescriptives. En termes architecturologiques, le Pattern n°61 prescrit ainsi pour la *dimension* « largeur », une mesure de 45 à 60 pieds dans le cadre d'une *échelle optique*. En effet l'*échelle optique* définit les opérations mentales qui impliquent de « *tenir compte d'un point de vue –au sens propre- pour donner à une partie ou au tout de l'espace architectural une mesure faisant intervenir les modalités suivant lesquelles elle va être vue* » (Boudon et al. 2000, p.175). Or la place est dimensionnée de manière à ne pas « paraître » trop vide quand on l'occupe (cf. Tableau 7 : « *they look and feel deserted* »). Les Patterns d'Alexander semblent regrouper des mesures ou ensemble de mesures de l'architecture dont la *pertinence* est déjà définie.

#### 4.1.1.2 L'appropriation de la notion de Pattern en informatique

La notion de Pattern construite par Alexander a été empruntée par d'autres champs, dont celui de l'informatique, où des « Patterns languages » ont été développés pour assister la conception de programme. Dans l'ouvrage de 1994 *Design Patterns: Elements of Reusable Object-Oriented Software*, « The Gang of Four » (constitué de Erich Gamma, Richard Helm, Ralph Johnson et John Vlissides) propose de reprendre la notion de Patterns pour construire des solutions génériques à des problèmes récurrents dans la programmation de logiciels en programmation orientée objet (Gamma et al. 1993). Pour expliciter la reprise de la notion de Pattern par ces auteurs, nous nous appuyons sur leur article « Design patterns: Abstraction and reuse of object-oriented design », publié en 1993 dans le cadre de la Conférence ECOOP'93 - Object-Oriented Programming.

Comme pour Alexander, les Patterns sont pour Gamma *et al.* une aide à la conception. Les Patterns proposés par Gamma *et al.* expriment des structures de stratégie de conception de programmes (Gamma et al. 1993, p.407) et, plus précisément, d'architecture de programmes. Pour ces auteurs, les Patterns, dans le cadre de la conception de logiciels (Gamma et al. 1993, p.407) ont certaines caractéristiques notables :

- ils proposent un vocabulaire commun pour les concepteurs de logiciel, leur permettant ainsi de communiquer plus efficacement ;
- ils formalisent et rendent réutilisable un savoir issu de l'expérience ;
- ils réduisent le temps d'apprentissage d'une librairie de classes (les librairies de classes sont des outils regroupant des fonctionnalités déjà écrites, utilisables dans un nouveau programme), en étant identiques à différentes librairies,
- ils fournissent un cadre d'organisation et d'optimisation des programmes à concevoir.

Les Patterns de Gamma *et al.* sont de niveaux d'abstraction variables et règlent des problèmes variés. On peut donner comme exemple les Patterns « Adapter » ou « Observer ». Le premier

---

<sup>41</sup> «Each pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice. » (Alexander 1977).

transforme le protocole d'une classe pour le rendre conforme à un autre et, le second, synchronise et coordonne deux objets (Gamma et al. 1993, p.422). Ces deux Patterns visent à régler des problèmes techniques en apportant au concepteur des stratégies ayant fait leurs preuves en programmation orientée-objet.

On s'aperçoit ici de la proximité de la notion de Pattern telle qu'elle est reprise par Gamma *et al.*, avec la notion proposée par Alexander : ils apportent tous deux des solutions génériques à des problèmes récurrents issus d'un champ d'expertise particulier. Mais l'appropriation de la méthode des Patterns d'Alexander en informatique n'est pas littérale. Ainsi, si l'on en croit le retour fait par Alexander en 1999 lors de sa communication « The Origins of Pattern Theory: The Future of the Theory, and the Generation of a Living World », réalisée dans le cadre des conférences ACM (Conference on Object-Oriented Programs, Systems, Languages, and Applications), les sciences informatiques auraient pris quelques distances avec la notion de Pattern telle qu'il l'avait définie. Pour Alexander, il y a deux points importants qui n'ont pas été repris en informatique : 1- les Patterns d'Alexander sont moraux : ils visent une « bonne solution » au sens moral, ce que ne visent pas les Patterns du génie logiciel dont le but est de proposer des « solutions efficaces »<sup>42</sup> (Alexander 1999, p.75); et 2- les Patterns d'Alexander sont génératifs et visent à produire des tous cohérents, ce que ne visent pas explicitement les Patterns en informatique<sup>43</sup> (Alexander 1999).

Cette distinction est majeure. Selon le retour que fait Alexander sur cette appropriation de sa méthode, les Patterns en informatique sont principalement des modes de communication, c'est-à-dire une méthode pratique pour pointer et expliciter des problèmes récurrents en programmation (Alexander 1999, p.74)<sup>44</sup>, mais ils ne visent pas forcément à « *profondément changer la vie des gens* » (« *a deeper role in human life* ») (Alexander 1999, p.74) ce que revendique Alexander dans sa méthode des Patterns. Cette intention de « valeur morale » du Pattern Language semble pouvoir être rapprochée, en termes architecturologiques, des *pertinences* des solutions dont nous avons vu précédemment qu'elles pouvaient être déjà données.

#### 4.1.2 Patterns et modélisation paramétrique : la proposition de Robert Woodbury

Robert Woodbury et son équipe proposent, quant à eux, d'interroger la notion de Pattern pour un nouveau champ d'expertise: la conception de modèles paramétriques. Plus particulièrement, Woodbury propose d'assister la mise en œuvre et l'apprentissage de la modélisation paramétrique par les concepteurs architectes et ingénieurs (Woodbury et al. 2007, p.223).

Pour Woodbury, un modèle paramétrique peut être constitué de « parties » identifiables et quasiment autonomes, ce qui peut tout à fait se vérifier en lien avec les mécanismes *paramétriques*

---

<sup>42</sup> « I understand that the software patterns, insofar as they refer to objects and programs and so on, can make a program better. That isn't the same thing, because in that sentence "better" could mean merely technically efficient, not actually "good". » (Alexander 1999, p.75).

<sup>43</sup> « Have you done that in software pattern theory? Have you asked whether a particular system of patterns, taken as a system, will generate a coherent computer program? If so, I have not yet heard about that. » (Alexander 1999, p.75).

<sup>44</sup> « It allows you to write down good ideas about software design in a way that can be discussed, shared, modified and forth. So, it is a really useful vehicle of communication. » (Alexander 1999, p.74).

*fonctionnels* mis en œuvre par les modélisateurs paramétriques (cf. Chapitre 2). Ces « parties » ou sous-systèmes peuvent, pour Woodbury, répondre à des problèmes de modélisation récurrents. Les solutions les plus performantes à ces problèmes sont alors formalisables sous la forme de Patterns. En s'appuyant sur le travail d'Alexander, Woodbury définit un Pattern comme une solution générique à un problème défini (Woodbury 2010, p.185). Un pattern comprend la définition d'un problème, d'une solution et d'un contexte (Woodbury 2010, p.185).

Robert Woodbury associe l'usage des patterns à une stratégie de modélisation qu'il nomme : « Divide and Conquer » (Woodbury 2010, p.188). « Divide and Conquer » est une stratégie qui consiste à diviser un projet de modèle paramétrique en parties modélisables facilement. Cette organisation du projet ne se fait pas sur le modèleur mais s'esquisse sur un médium quelconque (Woodbury 2010, p.188)<sup>45</sup>. Une fois que ces sous-ensembles constituant le modèle sont identifiés, le concepteur cherchera le moyen de les modéliser le plus rapidement possible en rapprochant les sous-ensembles identiques et en faisant appel à des patterns. « Divide and conquer » est une stratégie visant à organiser un modèle paramétrique en sous systèmes identifiables et compréhensibles, aux interactions simples. La stratégie « divide and conquer » permet de rendre le modèle paramétrique plus adaptable aux transformations : les « parties » du modèle étant bien identifiées et autonomes, elles peuvent être transformées presque indépendamment. Voire même, une partie d'un modèle peut être pensée comme « inaboutie », l'isoler permet alors de différer une prise de décision du concepteur (Woodbury 2010, p.189). Les données d'entrées et celles de sorties de sous-systèmes doivent être explicitement clarifiées<sup>46</sup> : cela permet par la suite de transformer ces sous-groupes plus facilement.

Les patterns proposés par Woodbury sont plus proches de l'acceptation définie dans le champ de l'informatique par Gamma *et alii* que de l'acceptation définie dans le champ de l'architecture par Alexander. On observe dans le Tableau 8 une comparaison synthétique de ces trois acceptations.

---

<sup>45</sup> « Start with a sketch in any medium you wish; just make it quickly. Divide it into logical parts, so that each part can be modelled easily. » (Woodbury, 2010. p. 188).

<sup>46</sup> Dans le cas de la modélisation sur Grasshopper, l'utilisation de « receveur-émetteur » peut être un complément pratique de la méthode « divide and conquer ». Elle permet d'identifier les liens entre les différentes parties d'un modèle en les « temporisant » *via* des « params », des composants visant seulement à contenir des informations. Ces params permettent alors de visualiser les entrées, et de les transformer facilement en cas de besoin.



|                                     | <i>Patterns construits par Alexander</i>  | <i>Patterns construits par Gamma et alii</i>   | <i>Patterns construits par Woodbury</i>  |
|-------------------------------------|---|--|--|
| Problèmes assistés :                | Problèmes de mesure de l'espace   | Problèmes de conception de programmes  | Problèmes de mesure d'un modèle paramétrique   |
| Visées des patterns :               | Assister la conception (architecturale)   | Assister la conception de programmes orientés objet  | Assister la modélisation paramétrique pour les architectes   |
|                                     | - Aide à l'explicitation d'un « contexte » de conception (les variables d'inadaptation)<br>- « <i>profondément changer la vie des gens</i> » (Alexander 1999, p.74) | - Aide à la communication,<br>- aide à la formalisation de savoirs<br>- pointent et explicitent des problèmes récurrents | - Aide à la communication,<br>- aide à la formalisation de savoirs<br>- pointent et explicitent des problèmes récurrents |
| Utilisateur visé :                  | architectes   | informaticiens   | architectes  |
| Donne de la mesure ce qu'il assiste | (l'architecture)<br>OUI   | ---  | (les modèles paramétriques, l'architecture)<br>?*  |

**Tableau 8: Comparatif des acceptions de patterns par Alexander, Gamma et alii et Woodbury**

Il est important de noter que contrairement aux Patterns d'Alexander, les Patterns de Woodbury ne visent pas à assister la conception architecturale, mais ils visent à assister la conception de modèle paramétrique pour la conception architecturale. Les Patterns de Woodbury contiennent-ils des pertinences architecturales ou paramétriques prédéfinies (cf. \* Tableau 8) ? est une question soulevée dans l'analyse présentée dans le Chapitre 7.

Woodbury et son équipe ont construit un ensemble de Patterns spécifiques à la modélisation paramétrique à partir d'observations de cas, reprenant ainsi la méthode proposée par Alexander et son équipe (cf. Figure 48) (Arnold 1977, p.35). Ces recherches sont documentées par des observations-participantes auprès de concepteurs, autant architectes qu'ingénieurs (Marques & Woodbury 2006; Marques & Woodbury 2007; Qian et al. 2007; Woodbury 2010; R. Aish & Woodbury 2005). La méthode suivie est une recherche de récurrences dans les modèles utilisés par les concepteurs observés, ainsi que dans les questions soulevées par ces concepteurs dans leurs pratiques de la modélisation paramétrique (Qian et al. 2007).

Treize Patterns sont construits à ce jour (cf. Tableau 9). Ils constituent cinq catégories :

1- Une catégorie de Patterns visant à rendre un modèle clair et lisible, cette catégorie contient le Pattern clear name<sup>47</sup> ;

2- Une catégorie de Patterns visant à résoudre des problèmes liés au contrôle d'une partie du modèle au moyen d'une autre partie, ce sont des modèles de techniques de structuration (Woodbury, 2010. p.187), dont les Patterns : controler, jig, increment et reactor font partie;

3- Une catégorie de Patterns visant à spécifier et positionner des objets (Woodbury, 2010. p.187), comprenant les Patterns : points collection et place holder;

4- Une catégorie qui vise à extraire des données spécifiques à partir d'un modèle existant (Woodbury, 2010. p.187), ces données pouvant être abstraites, évaluées, sélectionnées, *etc.*, cette catégorie comprend les Patterns : projection, reporter et selector ;

5- et enfin, une catégorie (Woodbury, 2010. p.187), comprenant les Patterns : mapping, recursion et goal seeker.

Les Patterns développés par Woodbury et son équipe sont présentés ci-dessous de façon succincte.

---

<sup>47</sup> Nous reprenons la convention utilisée par Woodbury dans son ouvrage *Elements of parametric design*, et distinguons les noms de patterns en les mettant en page avec la police Verdana.

| <i>Nom du pattern</i>   | <i>Rapide description de la fonction du Pattern</i>  |
|-------------------------|--|
| <b>clear name</b>       | Utiliser des noms clairs et signifiants pour désigner les objets   |
| <b>contrôler</b>        | Contrôler (une partie d') un modèle au travers d'une autre partie de modèle<br><i>Exemple</i> : contrôler la longueur d'une ligne en fonction de son éloignement à un point, contrôler une direction à partir d'un azimut et d'une altitude  |
| <b>jig</b>              | Construire un cadre simple et générique pour isoler la structure et la position d'un détail géométrique<br><i>Exemple</i> : contrôler la variation d'une surface en l'incluant dans un parallélogramme fictif permettant de maîtriser les transformations  |
| <b>increment</b>        | Diriger une transformation grâce à une série de valeurs relatives<br><i>Exemple</i> : diriger la hauteur d'une hélice grâce à une valeur   |
| <b>point collection</b> | Organiser une collection de points comme un objet composé de la répétition d'éléments localisés<br><i>Exemple</i> : Positionner une séquence de points sur une courbe paramétrique   |
| <b>place holder</b>     | Utiliser des objets « proxy » ( ? ) pour organiser des collections complexes<br><i>Exemple</i> : positionner/répéter un module sur une trame de points   |
| <b>projection</b>       | Transformer un objet dans un autre contexte géométrique<br><i>Exemple</i> : faire une symétrie, une projection centrale sur un plan  |
| <b>reactor</b>          | Faire répondre un objet à la proximité d'un autre élément<br><i>Exemple</i> : faire réagir le tracé d'une courbe à la proximité d'un point   |
| <b>reporter</b>         | Extraire des informations d'un modèle : les re-présenter sous une autre forme<br><i>Exemple</i> : extraire les propriétés d'une collection de point organisé en tableau, extraire un fichier Excel avec les courbures moyennes associées à un groupe de surfaces, faire une symétrie axiale, introduire des informations de couleur dans le modèle relatif à une propriété géométrique |
| <b>selector</b>         | Sélectionner des éléments d'une collection qui ont des propriétés spécifiques<br><i>Exemple</i> : sélectionner des objets en fonction de leur distance par rapport à un point  |
| <b>mapping</b>          | Utiliser une fonction dans un nouveau domaine  |
| <b>recursion</b>        | Répliquer un motif de façon récursive<br><i>Exemple</i> : le tapis de Sierpinsky   |

|                                   |  |
|-----------------------------------|--|
|                                   |  |
| <b>goal seeker</b>                | Transformer une entrée (input) jusqu'à ce qu'une sortie (output) franchisse un certain seuil<br><i>Exemple</i> : modifier une courbe jusqu'à ce qu'elle atteigne un point spécifique |
| <b>Recorder (M.R. S.)</b>         | Enregistrer sélectivement les choix définis par le concepteur lors de l'exploration d'instances  |
| <b>Hysterical State (M.R. S.)</b> | Créer de nouvelles variations à partir de la recombinaison de décisions précédentes enregistrées comme paramètres (par le pattern recorder)  |

**Tableau 9: Patterns proposés par Robert Woodbury et son équipe dans *Elements of Parametric design* (2010)**

Pour Woodbury et son équipe, la méthode des Patterns serait une aide à la modélisation paramétrique, entre autres, grâce à quatre attributs:

- Les Patterns sont *explicités*: ils accompagnent la réflexion du concepteur en le forçant à clarifier ses intentions. De plus, l'usage de Patterns peut aider un groupe à se construire une bibliothèque de modèles et cette clarification des idées favorise leurs diffusions et leurs communications.

- Les Patterns sont *partiels*<sup>48</sup> : ils sont plus qu'un simple nœud, mais ne sont pas aussi complexes qu'un modèle. Les Patterns ne sont que des fragments de stratégies qui doivent être composés lors de la conception d'un modèle. Cette caractéristique des Patterns permet au concepteur de formaliser un modèle en modules identifiables.

- Les Patterns *répondent à des problèmes précis* et partagés<sup>49</sup>.

- Les Patterns proposent des *solutions abstraites*, génériques<sup>50</sup>.

Ces patterns sont sous-tendus par une acception de la conception architecturale comme activité à la fois de transformations (« *change* » (Woodbury 2010, p.7)) et de résolution de problèmes (Qian et al. 2007). Dans *Elements of Parametric Design*, R. Woodbury (2010) avance que la modélisation paramétrique est particulièrement pertinente pour la conception car elle permet de « représenter le changement »<sup>51</sup> : c'est-à-dire l'exploration de multiples variations (*cf.* Chapitre 2). Il en conclut une affinité entre la conception et la modélisation paramétrique : « *Design is change. Parametric modeling represents change.* » (Woodbury 2010, p.7). Nous avons vu dans le Chapitre 2 à propos des mécanismes informatiques de la modélisation paramétrique que, si ceux-ci permettent en effet de représenter plusieurs variantes d'une géométrie, la modification d'un modèle est contrainte par la chaîne de dépendances existante.

Un autre argument avancé par Woodbury à propos de la pertinence des Patterns pour la pratique de l'architecture procède de leur origine : les Patterns résultant d'observations de récurrences de problèmes

<sup>48</sup> « *partial* (above nodes and below designs) » (Woodbury 2010, p.188).

<sup>49</sup> « *problem-focused* (shared problems) » (Woodbury 2010, p.188).

<sup>50</sup> « *abstract* (generic) » (Woodbury 2010, p.188).

<sup>51</sup> « *Parametric modeling represents change.* » (Woodbury 2010, p.7)

techniques dans la conception de modèles paramétriques dans la pratique d'architectes<sup>52</sup>, ces patterns seraient pertinents car utilisables et utilisés. Or, si les Patterns sont utilisés par certains architectes, seront-ils pertinents pour d'autres architectes dans le cadre d'autres projets ? Pour Woodbury, les patterns sont normatifs : les solutions qu'ils apportent sont associées à une doctrine. En effet ils sont impératifs, ils décrivent ce qui doit être fait (Woodbury 2010, p.186). Mais les solutions techniques que les patterns proposent sont-elles pertinentes pour l'aide à la modélisation paramétrique en conception architecturale ?

Une analyse architecturologique approfondie de l'usage de la méthode des Patterns pour la conception de modèles paramétriques en conception architecturale permet de pointer plus précisément les apports de cette méthode (cf. Chapitre 7). Dans quelle mesure une normativité de la technique de conception de modèles influe la conception architecturale quand celle-ci lui est conjointe ? Est une des questions soulevées par l'analyse proposée dans le Chapitre 7.

### **- Les samples : applications des patterns**

Ces Patterns sont présentés dans l'ouvrage de Woodbury *Elements of parametric design* publié en 2010, ainsi que sur le site [designpatterns]. Sur ce site, des exemples d'application de Patterns, appelés « samples » (Woodbury 2010; Woodbury et al. 2007), sont développés pour le modèleur Generative Components. Ces exemples ont également été développés sur Grasshopper et mis en ligne en 2009 par Tsung-Hsien Wang et Ramesh Krishnamurti de l'École d'Architecture de Pittsburgh [designpatterns.GH]. Il semble important de préciser que l'utilisation de Patterns relève de « stratégies de modélisation », tout comme les Patterns d'Alexander préconisaient des stratégies de conception applicables de manière toujours différente (Alexander 1977). Or Woodbury plébiscite l'utilisation de stratégies « copier-coller » qu'il associe aux Patterns (Woodbury 2010, p.37). Cela pose question : un pattern étant une stratégie générique pour la conception de modèles paramétriques, une telle stratégie générique peut-elle s'accommoder d'un fragment de modèle paramétrique déjà conçu ? Et comment constituer ces fragments de modèles réutilisables ? Ces fragments réutilisables pourraient être rapprochés des « Samples », par ailleurs également proposés par Woodbury.

Le terme « sample » est utilisé dans le champ de la composition musicale pour désigner un « échantillon » (extrait de musique ou son) pouvant être réutilisé en dehors de son contexte d'origine. Le terme « Sample » est couramment utilisé par les acteurs de la modélisation paramétrique comme nous pourrions le voir dans la suite de ce Chapitre. Nous utiliserons ici ce terme dans une acception proche de celui défini dans le champ de la musique : celle d'échantillon ou de module visant à être utilisé, modifié et/ou recomposé dans un nouvel objet.

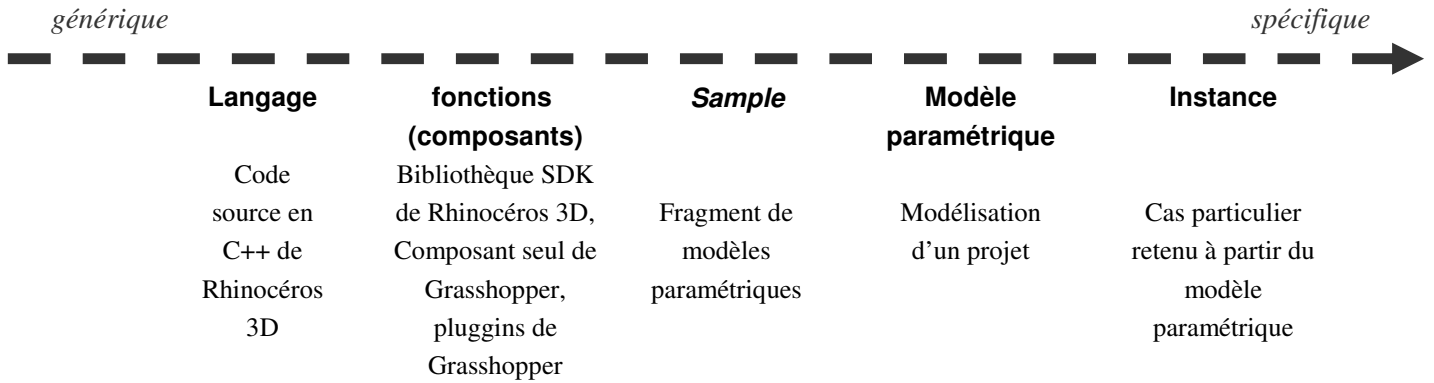
Pour chaque Pattern, Woodbury propose des « samples » visant à illustrer les stratégies proposées. Un sample est un fragment de modèle mettant en œuvre une stratégie proposée par un pattern. C'est une forme de pattern appliqué. Un sample est un appareillage de plusieurs fonctions composant un fragment de modèle. Cet appareillage peut être utilisé au sein d'un modèle paramétrique plus complexe et spécifique à un projet.

Si l'on situe les différents éléments manipulés lors de la modélisation paramétrique sur une échelle du plus générique au plus spécifique (cf. Figure 49), on observe alors une gradation allant des langages de programmation, avec lesquels un très grand champ de possible est ouvert pour qui sait les manipuler,

---

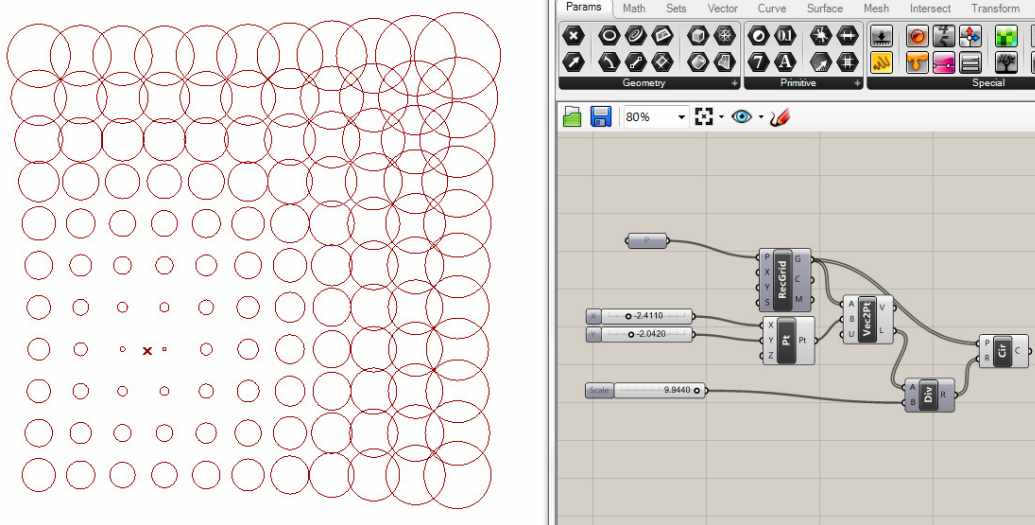
<sup>52</sup> « *Patterns work when grounded in practice.* » (Woodbury 2010, p.7)

au modèle paramétrique propre à un projet et définissant un domaine de solutions, pour aller jusqu'à l'instance : formes géométriques spécifiques à un modèle paramétrique et à une définition de ses paramètres. Un « Sample » se situerait entre la fonction (ou le feature logiciel) et le modèle paramétrique.



**Figure 49 : Proposition de conceptualisation des granularités de la modélisation paramétrique, du langage aux instances géométriques, exemplification pour le cas de Grasshopper et Rhinocéros 3D**

Par exemple le Sample « ReacCirclesRadius » [designpatterns] est un exemple d'application du Pattern Reactor. Le Pattern Reactor permet de contrôler un élément du modèle par un objet autre. Dans le sample ReacCirclesRadius, les rayons d'une trame de cercles sont contrôlés par la distance de chacun des cercles à un point. Comme on le voit dans la représentation *explicite* de ce sample, à gauche de la Figure 50, plus les cercles sont proches du point, plus leur rayon est réduit. Le sample est ici composé de plusieurs fonctions (les composants Grasshopper, visibles dans la représentation *symbolique* du sample, à droite de la Figure 50) et, on peut imaginer l'utilisation de ce sample dans un modèle paramétrique plus complexe, comme par exemple pour modéliser une façade aux ouvertures multiples et irrégulières dont on voudrait contrôler la porosité par rapport à des points remarquables (vues à ménager, entrées de lumière, etc.).



**Figure 50 : "ReacCirclesRadius", à partir de [designpatterns.GH]**

Samples et Patterns sont donc ici bien différents : là où le Pattern propose une stratégie de conception de modélisation paramétrique (pour aider à générer une géométrie, à structurer un modèle, à extraire des données, etc.), le sample est une application spécifique d'une stratégie.

Les Samples permettent d'opérer des « copier-coller » de solutions techniques. L'aide à la conception de modèles paramétriques qu'ils proposent complète donc les stratégies génériques proposées par les patterns. Un sample est un medium entre un pattern et un modèle paramétrique.

Pour Robert Woodbury, la conception de samples est possible par tous, elle ne nécessite pas de recherche scientifique. Pour Robert Woodbury, un sample est une application de pattern dans le cas d'un projet particulier et, une formalisation du fragment de modèle produit pour le rendre lisible par d'autres. La lisibilité du modèle est un point qui peut paraître trivial mais qui est indispensable à la viabilité du modèle. En effet un modèle dont les éléments sont bien nommés (Woodbury 2010), correctement groupés et identifiés (Davis et al. 2011) peut facilement être transformé ou réutilisé. Un modèle paramétrique doit donc être synthétisé, réécrit, clarifié, même s'il fonctionne déjà, pour pouvoir être réutilisé. Organiser un modèle paramétrique nécessite: - de structurer un modèle en parties logiques clairement identifiables, - de synthétiser et d'optimiser *a posteriori* le modèle lui-même, -de rassembler sous la forme de « modules » les parties répétitives du modèle (par exemple via la fonction cluster sur Grasshopper), -nommer correctement les différentes variables utilisées, -commenter le modèle paramétrique. Les samples produits ainsi peuvent être partagés avec la communauté de la modélisation paramétrique (Woodbury 2010, p.189). Selon Woodbury, la conception de samples permet à un concepteur de clarifier sa réflexion et d'augmenter ses compétences professionnelles et ses performances en évitant la répétition des tâches. Ce travail intervient dans une phase avancée de la conception du modèle paramétrique, comme un retour réflexif sur le modèle produit, afin de rendre le modèle plus lisible, plus efficace et finalement réutilisable et partageable. Pour Woodbury (Woodbury 2010, p.189), développer ses propres outils est le niveau le plus haut de maîtrise de la modélisation paramétrique. Pour Woodbury, les architectes préfèrent reconstruire un modèle paramétrique plutôt que de réutiliser un modèle existant (Woodbury 2010, p.36), la réutilisation étant pourtant pour lui plus efficace : cela évite de faire plusieurs fois de suite le même travail de modélisation.

La méthode des patterns proposée par Robert Woodbury paraît donc pouvoir assister la conception de modèles paramétriques en architecture, en particulier grâce aux recours à des samples permettant de médiatiser les stratégies génériques et les modèles paramétriques spécifiques à une conception architecturale. Si le recours aux samples semble rendre particulièrement accessible la méthode des patterns, Woodbury ne propose que peu de samples en liaison avec les patterns construits. Par contre de nombreux samples sont diffusés sur internet.

Quels sont ces samples diffusés sur internet ? De quelles aides à la conception de modèles paramétriques sont-ils accompagnés ? sont les questions soulevées par la suite du Chapitre.

Comment de mêmes samples sont-ils utilisés pour accompagner des processus de conception architecturale différents? est une questions soulevées par le Chapitre 7.

## **4.2 Supports en ligne : informations en libre accès**

Outre Woodbury et quelques autres chercheurs universitaires (Davis et al. 2011), d'autres acteurs de l'architecture (étudiants, architectes, professionnels spécialisés dans le paramétrique, etc.) s'intéressent activement à l'aide à la modélisation paramétrique. Les aides produites par ces acteurs sont divers (articles, tutoriels, plugins pour modeleurs paramétriques, etc.). Nous nous intéressons dans cette thèse aux aides publiés sur internet. En effet, celles-ci sont susceptibles de toucher un grand nombre de concepteurs. L'intérêt constaté de ces aides et leur portée semblent suffisamment conséquente pour que nous y focalisions notre analyse.

Dans un premier temps, nous nous intéressons à la diffusion sur internet de fragments de modèles (appelés samples) structurés sous la forme de catalogues (4.2.2). Puis nous nous intéressons plus largement aux différents supports à l'usage de la modélisation paramétrique disponibles sur internet (4.2.1). La diffusion sur internet de ces ressources n'est pas neutre : elle implique des effets de communautés pour l'entraide et le partage de ressources, objets étudiés dans le paragraphe (4.2.3).

### **4.2.1 Les supports en ligne**

Nous l'avons vu, les mécanismes paramétriques demandent aux concepteurs d'explicitier une représentation symbolique (*cf.* Chapitre 2). Cette explicitation demande des connaissances spécifiques en informatique et en géométrie. Permettre l'accès à des ressources peut donc être considéré comme une aide à la modélisation paramétrique. Nous nous intéressons ici aux ressources, considérées largement (sans se restreindre aux modèles déjà développés), publiées sur internet. Pour aborder ces ressources, nous nous attachons ici à décrire les plateformes majeures de diffusion : 1- des plateformes « officielles » des éditeurs de logiciels paramétriques, 2- des plateformes développées dans le cadre d'enseignement de la modélisation paramétrique et, 3- des plateformes développées par des agences ou des initiatives individuelles.

#### **4.2.1.1 Supports en ligne entretenus par des éditeurs de logiciels**

Nous décrivons ici les sites « officiels » des trois principaux logiciels étudiés dans cette thèse : Grasshopper, Generative Components et Digital Project.

##### **- [grasshopper3d.com] supporté par McNeel pour Grasshopper**

Le support développé par McNeel pour Grasshopper est le site [grasshopper3d.com]. Ce site est appelé lorsque, sur Grasshopper, l'utilisateur sollicite la rubrique « aide » dans le menu « ? ».

Sur cette plateforme, la plupart des contenus ne sont pas produits par les équipes financées par McNeel, mais par les utilisateurs eux-mêmes. McNeel semble solliciter la communauté des utilisateurs de Grasshopper en fournissant divers services (forum, tutoriels, etc.) et espaces de visibilité en ligne pour les utilisateurs de grasshopper eux-mêmes (forum, pages personnels, galeries d'images, etc.). Sur le forum de [grasshopper3d.com], on retrouve le service d'une équipe de « debugging » des problèmes et difficultés exprimés par les utilisateurs. Ce service est prodigué par Giulio Piacentino et David Rutten (McNeel), mais il peut également être proposé par de « simples » utilisateurs.

Le site [grasshopper3d.com] se compose ainsi:

- D'espaces personnels de valorisation de ses « membres » :



Un espace personnel peut être créé par n'importe quel utilisateur : c'est le « profil ». Il est créé par chaque membre et regroupe des informations (relevé des actions, images publiées, etc.) rassemblées dans « my page » et visualisables en ligne. Cela attribue à [grasshopper3d] un rôle de réseau social. Si une grande visibilité des utilisateurs est donnée dès l'accueil du site, certaines informations ne sont accessibles que par les « membres » connectés.

- D'une organisation de membres par « groupes »:

Les groupes rassemblent des membres autour de projets spécifiques. La plupart du temps ce sont des projets de développement de plugins pour Grasshopper, tels que « Kangaroo », qui permet d'intégrer des modules d'analyse structurelle ou, « Firefly » qui intègre des fonctions pour l'interopérabilité de Grasshopper avec Processing<sup>53</sup>, des flux rss, etc. Ces plugins, développés par des utilisateurs, ont parfois été, à long terme, intégrés à Grasshopper, comme Kangaroo et Firefly.

- Des contenus publiés par la communauté :

Le site est composé de trois rubriques principales : « galleries » « tutorials » et « events ». La page « Galleries » rassemble les images et les vidéos (mais pas de modèles) de projets ou d'expérimentations, mis en ligne par les membres. La page « Tutorials » regroupe les tutoriels de David Rutten (McNeel) et renvoie vers des tutoriels développés par d'autres acteurs. La page « events » rassemble les annonces d'événements se rapportant à Grasshopper : séminaires, colloques, ateliers, etc. Ces événements ne sont que rarement organisés par l'entreprise McNeel elle-même.

- D'un lieu d'échanges directs :

Le cœur du site est le « forum » où se retrouvent les utilisateurs, experts ou non, et les développeurs de Grasshopper (en particulier Rutten, très présent dans les conversations du forum). Tous les membres peuvent poster des questions ou des sujets de discussion sur le forum et, de la même manière, tous les membres de la communauté peuvent répondre (cf. Conversations choisies, insérées au sein de l'annexe 5).

Toutes les discussions ne remportent pas un succès égal. Certaines requêtes peuvent ne pas être reprises par la communauté et rester sans réponse. En général, les requêtes ne rencontrant pas d'écho sur le forum sont déjà résolues dans des discussions précédentes ou font preuve d'une trop grande ignorance de la culture numérique en générale ou paramétrique en particulier. La réactivité ou l'absence de réactivité de la communauté face aux requêtes, semble faire ressortir les thèmes privilégiés de la culture de cette communauté. Dans une certaine mesure, ce forum fait preuve de « *modération communautaire* » (Casilli 2010, p.319): le groupe sélectionne les « bons » sujets, c'est à dire ceux qui méritent d'être débuggés et développés. Mais si la modération est communautaire, c'est-à-dire propre au groupe plutôt que propre à des modérateurs McNeel « officiels », l'importance des intervenants McNeel n'est pas à négliger. David Rutten et Giulio Piacentino répondent majoritairement aux requêtes du forum.

On peut observer, dans les discussions, que se sont souvent les mêmes acteurs qui répondent aux requêtes. Ces membres experts amènent des indications, corrigent les fichiers joints aux questions ou proposent des solutions aux requêtes. Ce support personnalisé en ligne est majeur pour la communauté Grasshopper. Une anecdote semble significative du poids de ce service : quand David Rutten prévoit de

---

<sup>53</sup> Processing est un environnement de programmation développé à l'origine pour enseigner la programmation, il est aujourd'hui utilisé par de nombreux acteurs, en particulier par des artistes [processing.org].

ne pas pouvoir assurer ce service pendant une certaine période, il prévient « officiellement » toute la communauté *via* un billet sur le forum.

Dans le forum, les discussions sont classées en quatre catégories : « errors and bugs », « VB C# et Python » et « Samples and Examples ». Dans « Samples and Examples », sont regroupées les discussions où les membres échangent des modèles pour les mettre à disposition de la communauté.

Le fonctionnement majoritaire de [grasshopper3D.com] par le forum permet une grande réactivité de McNeel face aux difficultés et aux bugs rencontrés par les utilisateurs de Grasshopper. De plus, cela rend la communauté active et dynamique et, crée des richesses en termes de ressources, par la soumission de modèles, l'échange d'expertises, le développement de plugins, l'émulation et la valorisation des membres.

#### - [gtwiki.org] de Gehry Technologies pour Digital Project

Gehry Technologies (*cf.* 2.3.3) propose comme support à l'usage de Digital Project, un « wiki » en ligne : [gtwiki.org]. Un wiki est une plateforme sur laquelle une communauté d'inscrits peut intervenir et publier ses ressources. Néanmoins sur [gtwiki.org] les interventions semblent peu nombreuses et la communauté n'est pas visible : il n'y a pas de visibilité des membres par profils, galeries, *etc.* et on ne connaît pas l'identité des contributeurs du site. Il est probable que la communauté rassemblée et pouvant publier des ressources sur [gtwiki.org] soit celle des experts salariés de Gehry Technologies.

Sur [gtwiki.org] deux types de ressources sont proposées : l'explicitation de notions et l'exposition d'exemples. Les notions sont exposées dans des « articles » sur des sujets précis, sous la forme de « FAQ », « tips » ou de tutoriaux. Ces articles sont organisés par ordre alphabétique, ce qui ne facilite pas toujours la recherche d'informations, puisque cela demande à l'utilisateur de savoir ce qu'il cherche. Le site comprend également des « Applications » et des « Reusable Contents » : les exemples. Ils guident la construction de modèles ou parties de modèles mettant en œuvre des notions précises (comme dans le cas de l'article : « *Changing a default set point positions for a beam* »). Ces articles de « cas » fournissent parfois des modèles déjà développés, comme dans le cas des « Knowledge Templates ».

#### - [bentley.com/gc] de Bentley System pour Generative Components

Sur [bentley.com/gc], Bentley System propose différentes ressources structurées et contrôlées pour l'usage de Generative Components. A la différence des ressources de [grasshopper3d] ou [gtwiki.org], tous les contenus de [bentley.com/gc] sont siglés et contrôlés. Les ressources de [bentley.com/gc] semblent construites par des professionnels. Là où les ressources de [grasshopper3d] relèvent quant à elles de la participation spontanée d'amateurs et de professionnels.

Les supports à l'usage de Generative Components proposés par [bentley.com/gc] sont :

- des cours en ligne prenant la forme de « salles de classe virtuelles » et de séminaires en ligne ;
- des ressources en ligne liées à la formation : comme des documents supports d'apprentissage (type manuel) et des démonstrations vidéos ;

- des exemples d'usage de Generative Components : des « case studies », présentés sous forme de pdf et de galeries d'images de projets (cf. Tableau 49).

La plupart de ces ressources ne sont accessibles que moyennant une inscription comme membre. Néanmoins [bentley.com/gc] n'est pas un réseau social : la connexion en tant que membre donne accès à des ressources spécifiques mais ne permet pas de publier de façon spontanée des informations sur soi ou sa production. La communauté des utilisateurs de Generative Components n'est pas visible sur ce support mais elle l'est sur une autre plateforme : celle de l'association « Smart Geometry », sponsorisée par Bentley System. « Smart Geometry » est une association qui n'est pas spécifiquement dédiée à l'usage de Generative Components mais qui s'intéresse à l'architecture numérique en général. Cette communauté est abordée plus précisément dans le paragraphe 4.2.3.

#### **4.2.1.2 Supports en ligne développés dans le cadre d'enseignements : filtres et aiguillages de contenus**

Aux supports officiels des modeleurs paramétriques, s'ajoutent des supports de ressources en ligne proposées dans des cadres pédagogiques d'enseignement de la modélisation paramétrique pour la conception architecturale. De plus en plus d'enseignements en école d'architecture semblent avoir recours à des sites web. Ils sont utilisés comme support de contenus pédagogiques et/ou de valorisation des productions des étudiants. Nous nous intéressons ici à des supports de cours visant à accompagner l'apprentissage de la modélisation paramétrique. La liste des sites présentée n'est pas exhaustive. Elle est néanmoins significative des enseignements institutionnels de la modélisation paramétrique en ligne.

La plupart du temps, les sites pédagogiques sont dédiés à un enseignement spécifique qu'ils décrivent et auquel ils donnent une identité numérique. Ces sites ne proposent que rarement de nouveaux contenus. Ils agglomèrent plutôt des liens vers des contenus existants, sélectionnés en vue d'aider le travail des étudiants. C'est le cas de sites comme [digital-knowledge.net] ou [studiolynn.at].

[digital-knowledge.net] est le site de l'Atelier de projet de Christian Girard et Philippe Morel de l'ENSA de Paris Malaquais. Ce site présente le cours, il décrit son contenu pédagogique au travers d'un texte, les enseignants et les conférences organisées. Il met également à disposition des étudiants une base de données. Cette base de données est constituée de liens renvoyant vers des contenus développés par [grasshopper3d.com] et le blog [ParaMod].

[studiolynn.at] est le site de l'atelier de projet de Greg Lynn à l'université des Arts Appliquées de Vienne. Ce site décrit l'enseignement (les cours, les projets précédents, les enseignants) et fournit quelques ressources au travers d'une liste de liens, en particulier des liens renvoyant vers des sites de praticiens.

Dans ces deux cas, la sélection d'informations et sa mise à disposition par les enseignants dans un lieu identifiable par les étudiants est, dans une certaine mesure, une aide à la modélisation paramétrique par l'aide à l'information. Ce mode d'aide à la modélisation paramétrique est le plus courant.

Dans d'autres cas, la sélection d'informations est également ouverte aux étudiants. C'est le cas de [architecturables.org] qui est le blog de l'enseignement de projet SP31 de Claire Petetin de l'ENSA de Versailles. Ce blog rassemble des billets postés par les intervenants du cours à partir de leurs ressources

ou de ressources publiées sur d'autres supports et des billets directement postés par les étudiants à partir de ressources trouvées sur internet ou d'éléments de leurs projets.

Le site [GHressources.blogspot.com] recense des contenus existants. Il s'inscrit dans une logique différente. Ce site est développé par Pablo C. Herrera de l'université de Sciences Appliquées du Pérou. Il n'est pas explicitement lié à un enseignement et aux objectifs pédagogiques d'une institution. Sur [GHressources.blogspot.com], Herrera recense de nombreuses ressources disponibles pour Grasshopper. Il renvoie vers des contenus variés : présentations de projets, tutoriels, bibliothèques de modèles, *etc.* Ce site a le mérite de rassembler des contenus fragmentés sur de nombreux supports, tels que des articles ponctuels de blog ou des pages de sites personnels.

La plupart des sites d'enseignement sont donc surtout des agglomérateurs de liens vers des contenus développés par d'autres. Ils ont un rôle de regroupement de données filtrées par les enseignants ou les étudiants eux-mêmes, en fonction des enjeux et des besoins des objectifs pédagogiques. Ces sites ont une réelle efficacité car ils permettent aux étudiants d'appréhender la diversité des ressources pour la modélisation paramétrique, par ailleurs fragmentées sur internet.

Certains sites font exception. C'est par exemple le cas de [co-de-it] vu précédemment mais également de [digitaltoolbox] développé par Tam Thien Tran et Scott Leinweber, qui diffuse de nombreux tutoriels. Cette plateforme n'est pas affiliée à une institution mais a été développée par ces deux acteurs qui se définissent comme des formateurs professionnels et des consultants.

Dans le cadre de ce travail de thèse, des supports de cours ont été conçus par François Guéna et moi-même pour l'apprentissage de la modélisation paramétrique sur Grasshopper. Ces supports seront décrits et analysés dans le Chapitre 8, où est présentée une proposition pédagogique d'enseignement de la modélisation paramétrique.

#### **4.2.1.3 Supports en ligne entretenus par des agences (ingénierie ou architecture) ou par des individus (consultants / enseignants / architectes... )**

Certaines agences (ingénierie, architecture, BET, ingénierie numérique...) cherchent à développer des ressources pour l'utilisation de modeleurs paramétriques. Ces sites se présentent souvent comme des « blogs d'agence », parfois développés en parallèle d'un site principal, à visée commerciale plus directe.

Dans la plupart des sites d'agences, mêmes spécialisées dans l'aide à la modélisation paramétrique, il est rare d'observer une réelle aide à l'usage de modeleurs. Ainsi les agences Evolute avec [evolute.at], Designtoproduction avec [Designtoproduction.com] ou encore Decode avec [decodebim.com] mettent à disposition leurs « books » et des publications relatives à leurs activités, mais pas d'aide à la modélisation paramétrique. Ce sont des sites à visée commerciale. Par contre, d'autres agences comme Hugh Dutton Associates avec [Complexity.com], Case consulting avec [designreform.net] ou encore l'agence Morphocode développent quant à elles, en plus d'un site à vocation commerciale, des sites internet où sont diffusés des ressources.

[Complexitys.com] est le blog de l'agence HDA, son éditeur est Francesco Cingolani. Ce blog ne propose pas de ressources informatiques, mais rassemble des articles courts sur une culture du numérique proche des préoccupations de l'agence et de Cingolani (Cingolani 2012)<sup>54</sup>. Le site [Designreform.net] développé par l'agence Case Consulting<sup>55</sup> propose quant à lui à la fois des articles (de culture numérique en générale ou plus spécifiques à des technologies), une galerie de projets et des tutoriaux pour divers logiciels (Maya, Paracloud, *etc.*) dont Digital Project et Grasshopper. Ces ressources sont systématiquement des tutoriaux décrits par des vidéos et de courts textes. Les modèles liés aux articles ne sont pas diffusés.

[Morphocode.com] est le site d'une agence d'architecture éponyme, sur lequel sont valorisées diverses productions : projets architecturaux, expérimentations formelles et expérimentations numériques. En particulier [Morphocode.com] met à disposition « Rabbit », un plugin pour Grasshopper, permettant de générer des LSystems. Divers articles sur des expérimentations sur Grasshopper sont aussi publiés avec les modèles liés. L'organisation chronologique du site ne paraît pas favoriser l'utilisation facile de ces ressources.

Ces sites permettent aux agences de développer leur visibilité comme experts de la modélisation paramétrique. Finalement, ces supports proposent comme aide à la modélisation paramétrique:

- des éléments théoriques permettant de constituer une culture numérique et paramétrique,
- des éléments de communication de cas résolus (avec ou sans diffusion des codes) et,
- parfois des éléments visant à construire de nouvelles fonctionnalités (plugins ou courts programmes).

D'autres sites sont développés sur des initiatives privées, sans financement extérieur. Les individus ayant développés ces sites sont souvent reliés à une institution (centre de recherche ou d'enseignement) ou à un milieu professionnel (agence d'architecture ou d'ingénierie). Ces sites proposent le plus souvent des contenus nouveaux.

Ainsi les chercheurs Karl Daubmann et Daniel Davis, dont l'objet d'étude est la modélisation paramétrique pour l'architecture, ont développé leur site internet: respectivement [Paramod.net] et [Parametricmodel.com].

[Paramod.net] est le blog de Karl Daubmann, professeur associé à l'université du Michigan et architecte de l'agence PLY Architecture. Sur ce blog, de courts articles sur des sujets variés sont publiés: des comptes-rendus d'enseignement, des communications sur des projets, et surtout des expérimentations sur les logiciels Grasshopper et Digital Project. Ces articles peuvent être considérés comme des ressources : ce sont des tutoriaux ou des démonstrations techniques. Ils associent la plupart du temps des explications (textes et images) aux fichiers finaux des modèles. Si la présentation

---

<sup>54</sup> Les apports plus larges du blogging dans la pratique d'HDA sont plus spécifiquement développés par Cingolani dans son article « Network Thinking et apprentissage social : vers une architecture en réseau 2/2 » (Cingolani 2012).

<sup>55</sup> L'origine de [designreform.net] n'est pas claire, car elle n'est explicitée par aucun texte d'« à propos » sur le site. Néanmoins la mise en valeur d'un lien vers « case consulting », le site d'une agence de consultant en ingénierie numérique de New York, porte à croire que les équipes sont les mêmes.

chronologique des articles ne facilite pas la recherche d'informations précises, l'organisation par mots clefs permet de cibler le domaine voulu.

D'autres blogs ou « pages personnelles » sont construits comme des sortes de « book » en ligne, où coordonnées et images de projets sont mises à disposition. Ces sites publient parfois des expériences sur des modeleurs paramétriques et peuvent alors mettre en ligne quelques codes téléchargeables.

Finalement, les supports en ligne d'aide à la modélisation paramétrique peuvent :

-Agglomérer, trier, diffuser des informations publiées ailleurs sur d'autres supports ;

-Diffuser des travaux en vue de valoriser une expertise paramétrique et en vue de la tester en la soumettant à d'autres utilisations ;

-Créer une communauté, partager un travail pour provoquer d'autres partages et stimuler une communauté.

On voit se dessiner ici des enjeux qui ne sont pas ceux usuellement impliqués dans les pratiques de l'architecture en agence, en particulier, l'enjeu de la constitution de communautés d'utilisateurs pour le partage d'expertises. Cet enjeu est questionné dans le paragraphe 4.2.3 suivant.

On observe finalement de nombreux supports à la modélisation paramétrique en ligne. Pour chacun de ces supports, on peut distinguer la nature des ressources diffusées (cf. Tableau 10).

|  | [Code Quotidien] | [Co-de-It] | [ParametricModel] | [Giulio Piacentino] | [gt-wiki] | [design Pattern.ca] | [design Pattern GH] | [grasshopper3D] | [bentley.com/gc] | [Smart Geometry.org] |
|--|------------------|------------|-------------------|---------------------|-----------|---------------------|---------------------|-----------------|------------------|----------------------|
| samples téléchargeables                        | oui              | oui        | oui               | oui                 | oui       | oui                 | oui                 | oui             | oui              | peu                  |
| Réponses personnalisées                        | non              | non        | non               | non                 | non       | non                 | non                 | oui             | non              | oui                  |
| Tutoriels, construction accompagnée de modèles | non              | oui        | non               | non                 | oui       | non                 | non                 | oui             | oui              | non                  |
| Connaissances explicitées, théoriques          | peu              | non        | non               | non                 | oui       | oui                 | oui                 | peu             | oui              | oui                  |
| Références de projets                          | peu              | non        | non               | non                 | non       | non                 | non                 | oui             | oui              | Oui                  |
| Organisation d'une                             | non              | non        | non               | non                 | non       | non                 | non                 | oui             | peu              | Oui                  |

|   |     |     |     |     |     |     |     |     |     |     |     |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| communauté en ligne (réseau social, visibilité des membres)         |     |     |     |     |     |     |     |     |     |     |     |
| Organisation d'une communauté physique (organisation de rencontres) | non | non | non | non | non | non | non | non | peu | non | oui |

**Tableau 10 : Ressources diffusées sur internet: synthèse des principales caractéristiques et des principaux services**

#### 4.2.2 « Catalogues » de samples en ligne

Dans l'aide à la modélisation paramétrique proposée par Woodbury, l'utilisation de codes « déjà développés » pour résoudre des tâches de modélisation autonomes peut être considérée comme une part de la méthode des patterns. En effet, ces samples sont une médiation entre la généralité des patterns et la spécificité des modèles paramétriques conçus en architecture. Ces samples sont nombreux sur le web. Nous en présentons ici quelques-uns.

Seuls deux sites diffusent des modèles exemplifiant explicitement des patterns. Ce sont les sites [designpattern], construit par Woodbury, et [designpatterns.GH], construit par un des centres de recherche en architecture de l'université de Pittsburg. Des modèles paramétriques non explicitement affiliés à des Patterns sont également diffusés. Ces modèles peuvent être diffusés ponctuellement, par exemple pour accompagner une démonstration dans un article ou un tutoriel (comme dans les sites [ParaMod] ou [MorphoCode]). Mais ils peuvent aussi faire l'objet d'une diffusion pensée sur le mode de la collection : ils s'inscrivent alors dans des catalogues mis à disposition sur des sites variés. Nous interrogeons ces catalogues dans ce paragraphe.

Parmi les sites proposant une aide à la modélisation paramétrique, nous présentons ceux dont les aides sont significatives. Ces supports et leurs spécificités sont décrits ci-dessous.

##### 4.2.2.1 Catalogues explicitement liés aux Patterns de Robert Woodbury

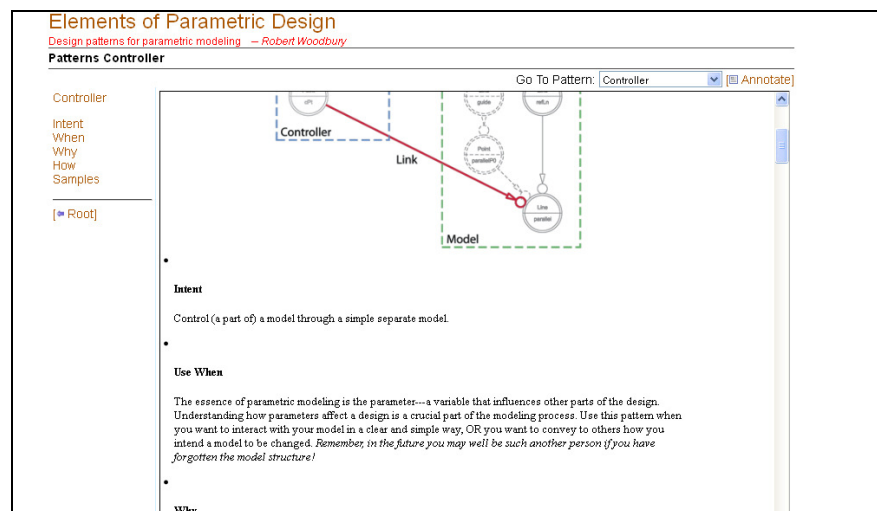
Deux sites sont explicitement liés aux Patterns développés par Woodbury : [designpattern.ca] et [designpattern.GH].

###### - [designpatterns]

Woodbury et son équipe proposent un site internet regroupant les Patterns identifiés dans le cadre de leurs recherches (cf. 4.1). Ce site, [designpatterns], reprend les principes de description et de structuration de Patterns construit à partir des travaux d'Alexander (Alexander 1977, p.xi) et de Robert Woodbury (Woodbury 2010, p.187). Chaque Pattern comprend :

- un nom clair et explicite ;
- un diagramme représentant la logique du Pattern de façon graphique ;
- Un paragraphe « What » décrivant le but du Pattern ;
- Un paragraphe « When » décrivant un scénario associant un problème et un contexte ;
- Un paragraphe « Why » décrivant les raisons d'utilisation du Pattern ;
- Un paragraphe « How » décrivant la façon dont le Pattern peut être utilisé pour résoudre un problème ;
- des Samples donnant des exemples d'utilisations du Pattern, les modèles de ces exemples sont téléchargeables ;
- Un paragraphe « Related Patterns » explicitant les connexions pouvant être pensées entre les différents Patterns, comme le recommande Alexander dans son ouvrage *Patterns Language* (Alexander 1977, p.xiii).

Ce site n'est autre qu'une version en ligne de l'ouvrage *Elements of Parametric Design* (cf. Figure 51).



**Figure 51 : Pattern controller, publié sur capture d'écran du site Elements of Pattern design (www.designpatterns.ca le 12-04-2012)**

Chaque Pattern est exemplifié par trois à quatre « Samples » développés sur Generative Components. Les Samples sont décrits par les images des géométries produites. Les fichiers des Samples peuvent être téléchargés.

- **[designpatterns.GH]**

[DesignPatternsGH] fait partie du portail de l'université Carnegie Mellon et a été développé par T.H. Wang. Le site diffuse, pour Grasshopper, des modèles reprenant les Samples proposés par Robert Woodbury pour illustrer ses Patterns. Cette collection reprend la catégorisation des Patterns et les textes diffusés par Woodbury sur [designpattern]. Pour chaque Sample, des images de la représentation symbolique et de représentations explicites des modèles sont données (cf. Tableau 11).



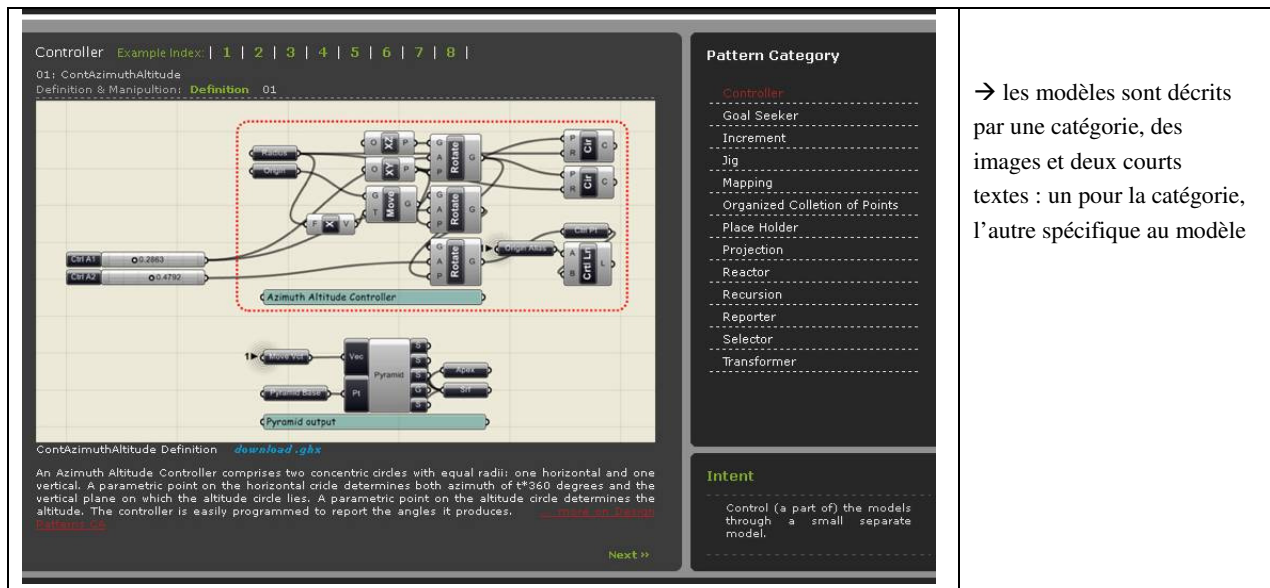


Tableau 11 : Capture d'écran de « design Patterns » (le 20-03-2012)

Ce site de diffusion, s'il reprend fidèlement les contenus construits par Woodbury, leur donne une forme qui semble plus adaptée au medium proposé par internet : celui-ci ne propose pas des « pages » figées et calquées sur un modèle papier, mais des « ressources » actualisables en fonction de la navigation de l'utilisateur (Monnin & Halpin 2012).

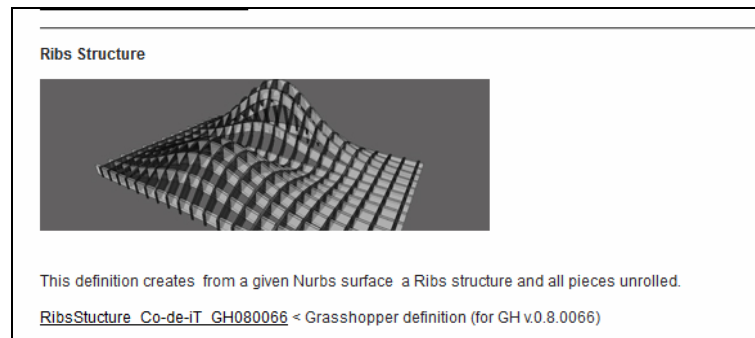
#### 4.2.2.2 Des catalogues de démonstrations techniques

Certains catalogues recensent des modèles qui permettent de réaliser des performances techniques. La description des modèles est plus ou moins développée. Si elle peut parfois comprendre une explicitation de l'enjeu technique et de la construction du modèle, comme dans le cas de [codequotidien], la plupart du temps elle se contente d'un titre et d'une très courte description constitués de mots clefs, comme dans [co-de-it], [giuliopiacentino] et [parametricmodels].

##### - [CodeQuotidien]

Dans [codequotidien], des modèles Grasshopper sont mis à disposition sous forme de listes organisées en fonction des tâches remplies (rotation, subdivision, etc.) (cf. Annexe 5 Tableau 46 p. 483). Chaque modèle est décrit de façon plus précise dans l'article lui correspondant. Pour le cas du modèle « Hybrid PQ » (cf. Tableau 12) de la catégorie de modèles « Discretization », l'article de description comprend deux applications possibles du code, ainsi qu'une visualisation du modèle Grasshopper, le modèle à télécharger, l'explicitation de la notion de discrétisation et la description du fonctionnement du code. Nous sommes proches ici d'une stratégie de Pattern, puisque contexte, problème et solution sont décrits. Cependant, comme dans tous les cas présentés dans ce chapitre, la solution proposée est spécifique aux cas développés. Finalement, [codequotidien] diffuse des « Samples » dont les Patterns ne sont pas explicités.





**Figure 52 : Modèle permettant de générer une structure Ribs, publié sur [Co-de-It] (capture d'écran du 09-12-2012)**

- **[Parametric Models]**

[parametricmodel] est un site développé par le chercheur Daniel Davis du laboratoire RMIT (Institut royal de technologie de Melbourne). Ce site est exclusivement dédié à la mise en ligne de modèles développés sur Grasshopper et Digital Project. Les modèles peuvent être proposés par tous mais sont modérés et sélectionnés par Daniel Davis. Des commentaires peuvent être ajoutés par les utilisateurs en lien avec les modèles. Les modèles sont présentés selon une liste chronologique d'implémentation du site. Des mots clefs, proposés par un menu déroulant, permettent de filtrer les cinquante-deux modèles. Ces mots clefs désignent les principaux objets (« params », « mesh », « vector », etc.) mis en œuvre dans les modèles. A cette description succincte des modèles sur la page d'accueil, des informations sont apportées dans des articles spécifiques (*cf.* Tableau 13). Les modèles eux-mêmes y sont décrits par les entrées et sorties impliquées, par les versions Grasshopper dont ils relèvent et par une description des tâches accomplies.

|  |  |
|--|--|
|  <p><b>Topographic Contours</b></p> <p>You can simply use this component for making topographic contour poly-lines. I have chosen mesh approximation to make it quick and efficient.</p> <p>Files for Grasshopper 0.8.x:<br/> <a href="#">Download</a> Topography_Contour.gh</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>B: Brep/Surface</li> <li>P: base plane - contours will be parallel to this plane.</li> <li>D: distance between contours</li> </ul> <p>Outputs:</p> <ul style="list-style-type: none"> <li>M: Converted mesh</li> <li>Crv: contour curves</li> <li>H(max): approximate maximum height of the surface</li> </ul> <p>Notes:<br/>Created by Pirouz Nourian</p> <hr/> <p>Component History:</p> <p>Viewed 1820 times<br/> Created by: Pirouz Nourian, 11-Jul-2011 07:22 pm<br/> Last edited by: @nzarchitecture, 10-Aug-2011 01:06 am<br/> <a href="#">View page history</a><br/> <a href="#">Edit this page</a></p> <hr/> <p>Comments (6)</p> <p>Stuart · 20 weeks ago<br/> I can't download the file. I get a string of text instead. Could you add a picture of</p> | <p>→ modèles décrits par une image, un titre et quelques mots.</p> <p>→ La version de Grasshopper est indiquée</p> <p>→ description des paramètres d'entrée du modèle</p> <p>→ description des données produites par le modèle</p> <p>→ L'historique indique les mises à jour mais aussi les auteurs du modèle.</p> <p>→ des commentaires sont possibles, ils sont propres au modèle. Ils semblent peu utilisés.</p> |
|--|--|

**Tableau 13 : Modèle permettant de générer des lignes topographiques, diffusé sur [parametricmodels] (capture d'écran du 20-03-2012)**

- [giuliopiacentino]

Giulio Piacentino est praticien et chercheur-formateur pour McNeel. Sur son site personnel [giuliopiacentino], il diffuse son book et des ressources qu’il construit pour Grasshopper. Il propose ainsi des modèles téléchargeables, des tutoriels et un pluggin de Grasshopper « WeaverBird ». WeaverBird permet de créer des maillages topologiques. Piacentino publie également de nouvelles fonctions pour Grasshopper, programmées en C#. Ces ressources sont présentées par catégories organisées en fonction des tâches accomplies. Pour chaque nouvelle fonction, sont décrites les entrées et les sorties, ainsi que la version pour laquelle elle est développée (cf. Tableau 14). Sont également proposées des images succinctes de représentations symboliques des modèles, mais pas d’images de leurs représentations explicites.

| BAKING & REFERENCING                        |   |  |
|---|---|--|
|   | <p>bakeName<br/> <a href="#">GH 0.6.0012</a><br/> <a href="#">GH 0.6.0018</a><br/> <a href="#">GH 0.6.0043</a></p>  | <p><b>obj</b> (geom) – a list of objects to bake<br/> <b>name</b> (string) – a list of names (common Rhino property)<br/> <b>activate</b> (bool) – one go-no-go signal, or a list thereof<br/> <b>A</b> (strings) – a list explaining the result(s)</p>  |
|   | <p>bakeAttributes<br/> <a href="#">GH 0.6.0012</a><br/> <a href="#">GH 0.6.0018</a><br/> <a href="#">GH 0.6.43</a><br/> <a href="#">GH 0.8.2</a><br/> <a href="#">GH 0.8.7</a><br/> <a href="#">GH 0.8.52 w/ groups</a></p> | <p><b>obj</b> (geom) – a list of objects to bake<br/> <b>name</b> (string) – a list of names<br/> <b>layer</b> (string) – Rhino layer name(s)<br/> <b>color</b> (onColor) – screen+print color(s)<br/> <b>lineT</b> (string) – the name of a linetype. Please only use with custom names.<br/> <b>pWidth</b> (double) – printing thickness(es)<br/> <b>wires</b> (int) – number(s) setting the quantity of wires, as in the property panel<br/> <b>material</b> (int) – GH shader(s), containing a Rhino material<br/> <b>activate</b> (bool) – one go-no-go signal, or a list thereof<br/> <b>A</b> (strings) – a list explaining the result(s)</p> |
|   | <p>findID<br/> findRef<br/> GH 0.6.0012</p>   | <p><b>x</b> (geom) – a Rhino geometry<br/> <b>A</b> (string) – The Rhino ID<br/> <b>A</b> (MRhinoObjRef) – The Rhino Ref</p>   |
| AUTOMATIC SECTIONS                          |   |  |
| <p><b>input</b> (geom) – the objects to</p> |   |  |

→ les modèles sont décrits par une catégorie, une image, un titre significatif et un texte de description des entrées et sorties du modèle

Les versions de Grasshopper sont indiquées

Tableau 14 : Fonctions développées d’instanciation vers Rhinocéros à partir de Grasshopper, publiées sur [giuliopiacentino] (capture d’écran du 20-03-2012)

### 4.2.2.3 Catalogues inscrits dans les sites d'aide à l'usage des modeleurs développés par leur éditeur

Des catalogues de modèles déjà conçus sont également diffusés sur les sites officiels d'aide à la modélisation paramétrique des modeleurs Grasshopper [grasshopper3d.com] et Digital Project [Gt-wiki].

- [grasshopper3d.com]

[grasshopper3d.com] semble être la plus importante ressource de « Samples » et « Exemples » de modèles Grasshopper sur le web. Ce site n'intègre néanmoins pas vraiment de « catalogue », les modèles sont publiés dans le cadre des forums du site (cf. Figure 236). L'accès privilégié à ces modèles reste donc les moteurs de recherche extérieurs au site.

- [Gt-wiki]

Sur [gt-wiki], des modèles nommés « knowledge templates » sont proposés pour l'usage de Digital Project. Ces « knowledge templates » sont de la « connaissance encapsulée » : c'est-à-dire de nouvelles fonctions rendues disponibles par le développement spécifique de fragments de modèles. Ces fragments de modèles sont souvent proposés sous la forme de « powercopy ». Une « powercopy » est une fonction proposée par CATIA, permettant de construire, à partir d'un modèle encapsulé, une « copie optimisée » utilisable comme un nouveau composant. Les « knowledge templates » sont décrits sur [gt-wiki] par des articles exposant les notions mises en œuvre (cf. Tableau 15).

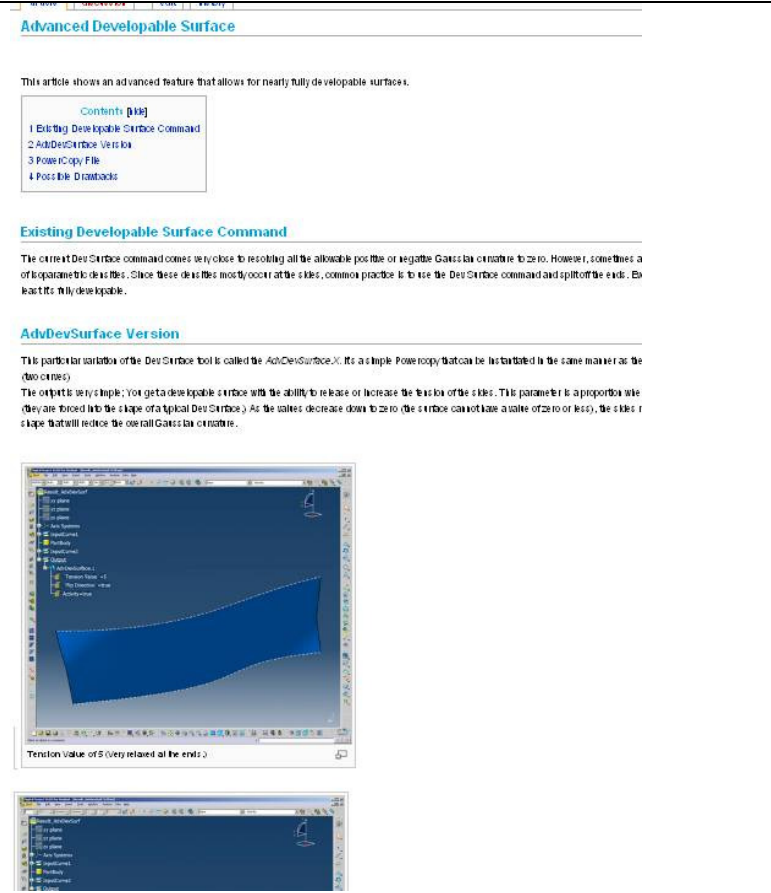
|   |   |
|---|---|
|  | <p>→ un article structuré pour présenter la notion encapsulée dans le Sample</p> <p>→ Description des moyens existant dans Digital Project pour répondre à l'objectif fixé par le Pattern</p> <p>→ proposition d'un nouveau moyen</p> <p>→ le fichier (sous forme de powercopy) est mis en ligne en fin d'article</p> |
|---|---|

Tableau 15 : Knowledge templates “AdvancedDevelopableSurface” publié sur [gt-wiki] (capture d'écran du 20-03-2012)

Cette revue de quelques catalogues de modèles publiés en ligne, pointe la variété des modèles diffusés. Ces modèles peuvent proposer de nouvelles fonctions (comme sur [giuliopiacentino] ou [gt-wiki]), des fragments de modèles paramétriques réutilisables (comme sur [parametricmodels], [co-de-it] ou [codequotidien]) ou encore des exemplifications de Patterns (comme dans [DesignPatterns] et [DesignPatterns.GH]). Ces modèles sont décrits et explicités avec plus ou moins de précisions (cf. Tableau 16), ce qui module l'intérêt de ces ressources.

Au sein de ces supports, la recherche d'un modèle ciblé est plus ou moins assistée. Ainsi, l'organisation, les différents regroupements des modèles ou l'utilisation de mots clefs peuvent favoriser ou non la recherche d'informations ou la sérendipité de l'utilisateur parmi les modèles (cf. Tableau 16 C).

|  | [CodeQuotidien] | [Co-de-It] | [ParametricModels] | [GiulioPiacentino] | [Gt-wiki] | [DesignPatterns.GH] | [DesignPatterns] |
|--|-----------------|------------|--------------------|--------------------|-----------|---------------------|------------------|
| <b>A/ Explicitation des notions mises en œuvre</b>                                     |                 |            |                    |                    |           |                     |                  |
| Description de la notion impliquée (intention, principe de géométrie ou programmation) | X               |            | X                  |                    | X         | X                   | X                |
| Description du fonctionnement du code  | X               |            | X                  |                    | X         |                     |                  |
| Description des applications possibles   | X               | X          |                    |                    | X         | X                   | X                |
| Explicitation de PATTERNS liés   |                 |            |                    |                    |           | X                   | X                |
| <b>B/ Caractéristiques générales</b>   |                 |            |                    |                    |           |                     |                  |
| Modeleur(s) concernés  | GH              | GH         | GH- DP             | GH                 | DP        | GH                  | GC               |
| Nombre de modèles en ligne   | 7               | 17         | 52                 | 14                 | ?         | 22                  | 22               |
| <b>C/ Descriptions et organisation des modèles</b>                                     |                 |            |                    |                    |           |                     |                  |
| modèles présentés en une liste   | X               | X          | X                  | X                  | X         |                     |                  |
| modèles organisés en catégories  | X               |            | X (tags)           | X                  |           | X                   |                  |
| Liste succinctes et description complète « à ouvrir »                                  | X               |            | X                  |                    |           | X                   |                  |
| Représentations symboliques directement lisibles                                       | X               |            |                    |                    |           | X                   |                  |

**Tableau 16: Explicitation des modèles diffusés et de leurs explicitations**

On observe donc que de nombreux modèles, ou fragments modèles, sont diffusés sur internet. Ces fragments de modèles peuvent être considérés comme des samples, même si les patterns dont ils relèvent ne sont pas explicités.

### **4.2.3 Présences en ligne et effets de communautés**

L'explicitation du paysage des ressources publiées en ligne permet de mettre au jour le rôle important de certaines communautés structurées autour de l'usage de modeleurs paramétriques. Ces communautés semblent accompagner et supporter ces pratiques.

#### **4.2.3.1 Importance des communautés**

Des réseaux d'individus pour l'échange d'informations, de compétences et de formations sur la modélisation paramétrique se mettent en place. Les profils de ces individus sont variés : il s'agit d'étudiants intéressés par l'architecture numérique, d'enseignants, de chercheurs, de développeurs employés par des éditeurs de logiciels, d'experts de la modélisation paramétrique travaillant comme consultants internes ou externes ou encore, d'architectes investis dans ces pratiques numériques. Ces communautés sont importantes car elles sont un vecteur de support et de diffusion des pratiques numériques (cf. 4.2.2 et 4.2.1). Nous l'avons vu, de façon flagrante dans le cas de [grasshopper3d.com] où les utilisateurs produisent la plupart des contenus du site.

Avec des modalités différentes, la communauté « Smart Geometry » est également une des communautés importantes de l'usage de la modélisation paramétrique. Cette communauté n'est pas spécifiquement dédiée à un modeleur, comme la communauté des membres de [grasshopper3d.com] l'est avec Grasshopper. Cette communauté est portée par un groupe d'individus moteurs dans le champ de l'architecture numérique, rassemblés en association. L'association « Smart Geometry » a en effet été fondée par Robert Aish, Hugh Whitehead, Jon Parrish et Lars Hesselgren en 2001. Ces fondateurs ont aujourd'hui des postes clefs dans l'industrie de l'architecture et de l'ingénierie numérique : Robert Aish dirige la cellule recherche et développement d'Autodesk. Hugh Whitehead est un des Partners de Foster and Partners et dirige le Specialist Modelling Group, cellule spécialisée dans l'usage de la modélisation paramétrique (cf. 3.3). Jon Parrish est le directeur de l'agence d'ingénierie ArupSport et Lars Hesselgren dirige la cellule de recherche et développement de l'agence d'architecture PLP. En 2006 Lars Hesselgren rend compte des objectifs de l'association dans un entretien avec Achim Menges. Pour lui, l'association vise en particulier l'enseignement et la diffusion d'une certaine culture du numérique et de la géométrie (Hesselgren in Menges 2006, p.44). De plus, la géométrie ainsi que l'optimisation (structurelle et environnementale) sont des préoccupations centrales pour les fondateurs de cette communauté, ce qui ne manque pas d'en colorer les orientations thématiques (Menges 2006, p.44). Une fois par an, l'association Smart Geometry organise trois jours de workshop et deux jours de conférences. La première rencontre a eu lieu en 2003 à Cambridge (GB). Cette communauté possède un support en ligne, [smartgeometry.org] composé d'une plateforme de diffusion de contenus accessible à tous, et d'un réseau social sur lequel les membres (les utilisateurs inscrits) partagent des images, des relations, des articles, un tweeter, *etc.*

Sur un autre plan, les communautés telles que celles rassemblées par les conférences CAAD Futures, eCAADe, ACADIA, SCAN etc. ont elles aussi leur importance. En effet, elles visent la mise en



commun de résultats de recherche et d'expérimentation dans la pratique ou la pédagogie de l'usage, entre autres, de la modélisation paramétrique.

Certaines des communautés dédiées à l'architecture numérique possèdent un support numérique et/ou des réseaux sociaux visant à faciliter les rencontres et les échangent d'informations. Ces sites proposent des fonctionnalités semblables à celles de facebook : avatar, profil avec possibilité de commentaires, groupes d'amis, discussions, partage de données, *etc.* (*cf.* le réseau proposé par [smartgeometry.org]). C'est le cas de [grasshopper3d.com] où le réseau social semble être un moyen pour les utilisateurs de promouvoir leurs productions et se tenir au courant des derniers développements, en particulier des développements des plugins structurés en « groupe ». C'est le cas aussi de [smartgeometry.org] où, cependant le réseau social paraît moins actif. On peut émettre l'hypothèse qu'il est plus utilisé lors des rencontres Smart Geometry, une fois par an.

Le partage de données, la valorisation et la visibilité des travaux semblent être certains des moteurs de ces processus d'échange.

#### **4.2.3.2 Ressources et interactions en ligne: Apports pour l'aide à la modélisation paramétrique**

L'organisation des utilisateurs de modeleurs paramétriques en communauté présente plusieurs intérêts pour l'aide à la modélisation paramétrique, dont :

- **L'implémentation de bases de données**

Les individus publient sur le réseau social leurs propres ressources (samples, articles, images, *etc.*). Le support numérique de la communauté (le site internet) devient alors un lieu rassemblant les productions de chacun. Ce lieu n'est pas un simple stockage : il permet la mise à l'épreuve de modèles pour la réutilisation, mais aussi la poursuite du développement de modèles par d'autres acteurs.

- **Le debugging**

Le rassemblement des individus en communauté permet le partage des expertises. Que ce soit un rassemblement virtuel (sur un forum) ou physique (lors d'un atelier) les individus plus avancés peuvent aider directement les plus débutants.

- **L'émulation, la valorisation**

Le rassemblement des données et des personnes permet non seulement de capitaliser des connaissances, mais également de motiver les individus en créant un sentiment d'appartenance à une communauté, un effet de reconnaissance et de valorisation pour les ressources apportées.

- **La veille informationnelle des agglomérateurs**

Les communautés créées autour de blogs ou autour de réseaux sociaux dédiés (facebook, twitter, grasshopper3d.com, *etc.*) sont des vecteurs de relais et de diffusion des ressources et des cultures nécessaires à la manipulation des modeleurs paramétriques.

Plusieurs aides à la modélisation paramétrique existent donc. Mais ces aides, pour la plupart techniques et génériques, sont-elles pertinentes pour l'usage de la modélisation paramétrique en conception architecturale ? La généralité des Patterns et la spécificité des Samples sont-elles pertinentes pour aider à concevoir un modèle paramétrique spécifique à un processus de projet architectural ?

Pour répondre à ces questions, la suite du document expose la recherche proposée dans le cadre de ce doctorat.

## **5 Construction de la méthode d'analyse et corpus rassemblé**

Cette thèse vise à produire des connaissances sur les opérations cognitives impliquées dans l'usage de modeleurs paramétriques pour la conception architecturale. Ces connaissances ces aides sont construites dans l'intention d'imaginer une pédagogie complémentaire aux aides existantes (cf. Chapitre 4). Le point de vue architecturologique est utilisé ici pour construire une connaissance des opérations cognitives de conception du modèle paramétrique et de l'architecture. Diverses analyses de cas a été menées. Ce chapitre explicite les méthodes d'analyse et de recueils de données mises en œuvre à cet effet, ainsi que le corpus effectivement rassemblé.

## **Plan du chapitre :**

|            |  |            |
|------------|--|------------|
| <b>5.1</b> | <b>Appareillage théorique et objet de recherche.....</b>   | <b>143</b> |
| 5.1.1      | Espace de la conception  | 143        |
| 5.1.2      | Opérations de la conception  | 145        |
| 5.1.3      | Conception architecturale et Conception de modèle paramétrique   | 147        |
| 5.1.4      | Les données recherchées  | 149        |
| 5.1.5      | Méthode de traitement des données  | 150        |
| 5.1.6      | Pour une objectivation de l'analyse  | 150        |
| 5.1.7      | Une méthode issue de l'Architecturologie Appliquée   | 151        |
| <b>5.2</b> | <b>Les entretiens.....</b>   | <b>154</b> |
| 5.2.1      | La méthode d'entretien   | 154        |
| 5.2.2      | Population interviewée et données recueillies  | 155        |
| <b>5.3</b> | <b>Les observations participantes.....</b>   | <b>166</b> |
| 5.3.1      | La méthode de l'observation participante   | 166        |
| 5.3.2      | Données issues des observations participantes  | 167        |
| <b>5.4</b> | <b>Les expérimentations .....</b>  | <b>170</b> |
| 5.4.2      | Données recueillies lors des expérimentations  | 177        |
| 5.4.3      | Spécificités du traitement des données des expérimentations  | 178        |
| <b>5.5</b> | <b>Synthèse du corpus .....</b>  | <b>181</b> |
| 5.5.1      | Exemple d'analyse architecturologique : le cas « Radical Production » de Daniela Kroenhert et Adam Orlinsky (Universität Die Angewandte de Vienne) | 185        |

## 5.1 Appareillage théorique et objet de recherche

L'objet scientifique de cette thèse est la caractérisation des opérations cognitives impliquées lors de la conception de modèles paramétriques en conception architecturale. Cette spécification vise à définir les connaissances et les mécanismes impliqués par ces processus, afin d'en proposer une pédagogie.

Pour appréhender les implications cognitives de l'usage de la modélisation paramétrique en conception architecturale, cette thèse convoque l'appareillage scientifique construit par l'architecturologie. L'architecturologie est un champ de recherche s'attachant à construire une connaissance de la conception et, en particulier, de la conception architecturale (cf. Chapitre 1.3). L'objet de connaissance de l'architecturologie est la conception comme *activité cognitive d'attribution de mesures*, interrogée en termes d'*opérations* de conception (Boudon 1971; Boudon 2004; Boudon 1991).

Dans ce paragraphe, nous rendons compte de l'appareillage théorique construit lors de la recherche pour répondre à la question « Quelles sont les opérations cognitives de conception impliquées lorsque la conception de modèles paramétriques est utilisée en conception architecturale ? ».

### 5.1.1 Espace de la conception

L'appareillage théorique de la thèse reprend le concept architecturologique « d'espace de la conception ». L'*espace de la conception*, tel qu'il est défini en 1971 par Philippe Boudon dans *Sur l'espace architectural* (nous utilisons ici l'édition augmentée de 2003), est un espace théorique désignant l'ensemble des opérations cognitives de la conception architecturale par lesquelles formes et mesures sont attribuées à un objet. L'*espace de la conception* permet de distinguer l'architecture comme édifice construit (que Focillon appelle l'espace « vrai », c'est-à-dire l'espace phénoménologiquement perceptible (Boudon 1971)) de son processus de conception (cf. Figure 53), qui constitue l'objet de connaissance de l'architecturologie.

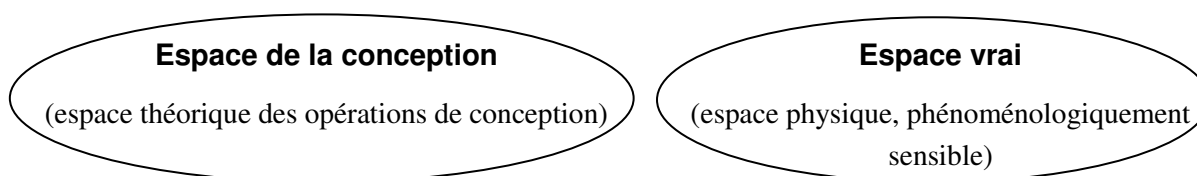
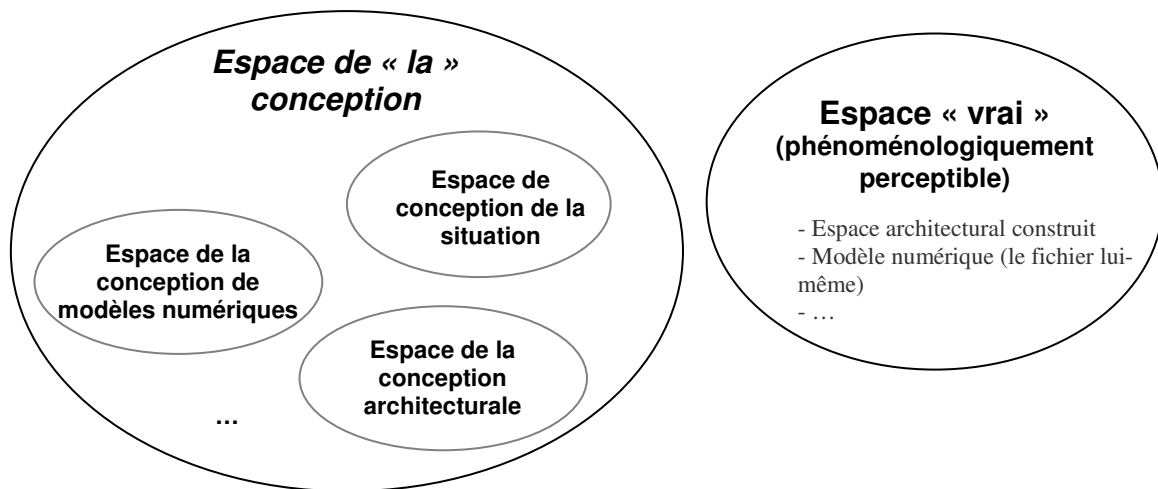


Figure 53 : Espace de la conception et espace vrai, d'après (Boudon 1971)

Philippe Boudon, dans son ouvrage *Conception* (2004), définit l'*espace de la conception* comme espace théorique regroupant les opérations de « la » conception, ne comprenant pas seulement les opérations de conception d'objets ou d'espaces physiques mais toutes les opérations de la conception entendues comme attribution de mesures, quel que soit l'objet de cette mesure.

Nous proposons d'interroger les « opérations de conception d'un modèle paramétrique » comme des opérations de « la » conception, visant l'attribution d'une mesure à un modèle paramétrique. Cette

présente thèse, vise donc à produire des connaissances sur *l'espace de la conception*, entendu comme espace des opérations d'attribution de mesures (cf. Figure 54), dans le cadre de l'usage de la modélisation paramétrique en conception architecturale.



**Figure 54: Schématisation possible de l'espace de la conception à partir de (Boudon 1971; Boudon 2004; Lecourtois 2006a; Ben Rajeb 2012)**

Les travaux du LAREA proposent également les concepts « d'espace de la représentation » (Boudon et al. 2000, p.79) et « d'espace de la modélisation numérique » (Lecourtois 2010). *L'espace de la représentation* désigne l'espace théorique des opérations cognitives de représentation d'un objet auquel on attribue des mesures. Les analyses exploratoires effectuées dans le cadre de cette recherche tendent à distinguer l'activité de modélisation paramétrique d'une activité de représentation. En effet, la modélisation paramétrique prend, la plupart du temps, la forme d'un travail de conception dans lequel la conception du modèle et la conception de l'architecture s'entremêlent.

L'« espace de la modélisation numérique » (Lecourtois 2010), quant à lui, est l'espace théorique des opérations de conception de modèles numériques. Ce dernier semble être un sous-espace de *l'espace de conception* (cf. Figure 54). Ce nouveau sous-espace ne paraît pas ici pertinent pour interroger nos objets qui sont les opérations de conception de modèles paramétriques et de l'architecture. Ces objets semblent plutôt se situer dans le plus général *espace de la conception*, sans nécessiter pour l'instant d'établir un « espace de la modélisation numérique ». Le rapprochement entre modélisation paramétrique et espace de la représentation et/ou espace de modélisation sera discuté. L'étude de *l'espace de la conception architecturale*, de *l'espace de la modélisation numérique* et des relations entre ceux-ci est un des axes de recherche du laboratoire MAP-maac (Lecourtois 2012, p.46)<sup>56</sup>, dans lequel s'insère ce présent travail de thèse.

<sup>56</sup> « Pour l'architecture numérique, la conception architecturale numérique serait l'activité cognitive complexe constituée d'une activité cognitive de conception architecturale, d'une activité cognitive de conception des modèles numériques et d'une activité cognitive de conception de liens d'influences de l'un sur l'autre. Ces liens d'influence peuvent être nommés « usages » de la modélisation numérique au sein de la conception et méritent de faire l'objet d'une approche pour en saisir leur conception en termes d'opérations – d'intégration, d'interférence, de dialogue, etc.-... de complexité. » (Lecourtois 2012, p.46).

## 5.1.2 Opérations de la conception

Par « opérations de la conception » sont entendues les opérations cognitives par lesquelles un concepteur donne de la mesure à un objet (Boudon 2002; Boudon et al. 2000; Boudon et al. 2005). L'architecturologie propose un appareillage de concepts construits de manière *a priori* pour décrire les mécanismes en jeu lors de cette activité d'attribution de mesures. Le concept majeur de l'architecturologie est celui d'*échelle*. L'« échelle » rend compte de la complexité opératoire de l'activité d'attribution de mesures. Dans cette présente recherche, nous proposons d'utiliser le concept de « dimension architecturologique » (c'est-à-dire ce sur quoi porte la mesure) pour appréhender les opérations de conception de l'objet architectural ainsi que les opérations de conception du modèle paramétrique.

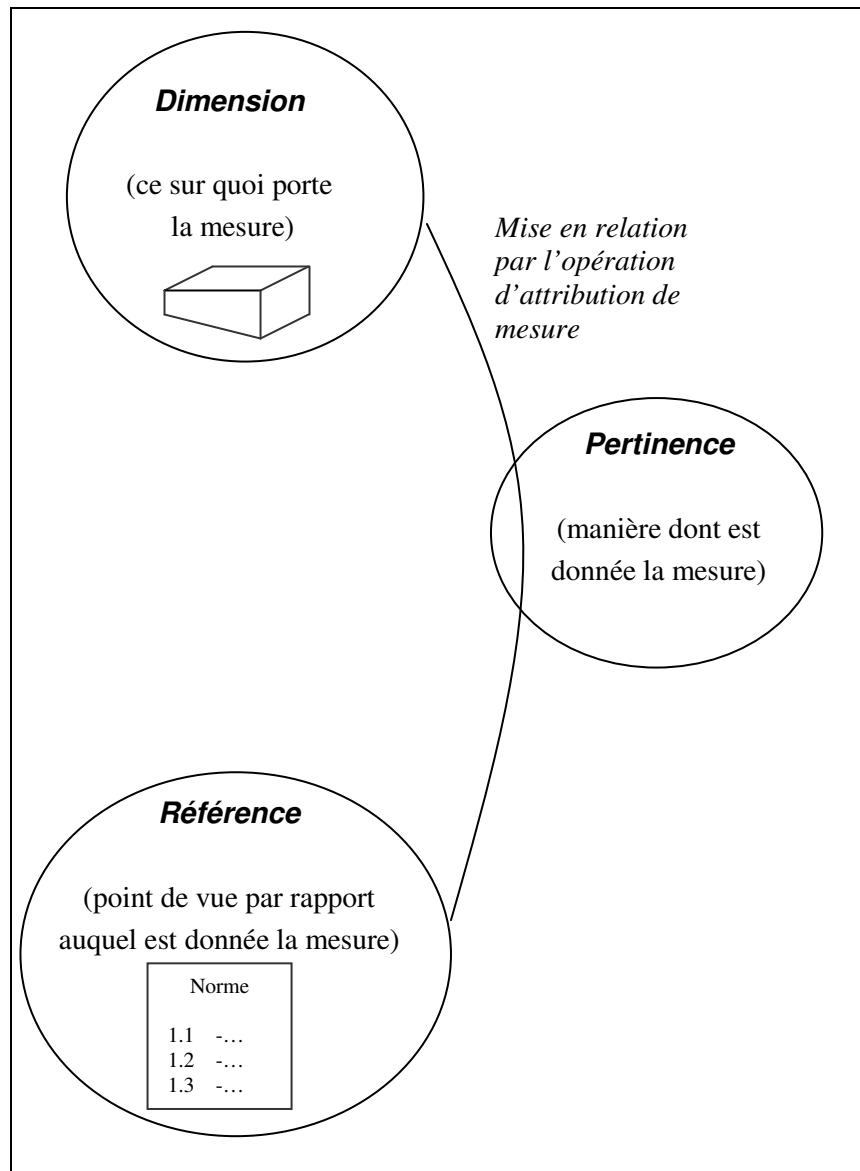
Le travail de conception, pris comme activité mentale par laquelle des mesures sont attribuées à un objet, est appréhendé comme un système associant *dimension*, *référence* et *pertinence*.

Les *dimensions* désignent ce à quoi est donnée la mesure (Boudon et al. 2000, p.124-128). Par exemple la longueur du bassin d'un projet de piscine devient une *dimension*, quand on décide de lui donner un métrage particulier.

Les *références* désignent ce par rapport à quoi est donnée la mesure (Boudon et al. 2000, p.109-113). Par exemple lorsque je décide de donner une valeur à la longueur d'une piscine en faisant référence à une norme olympique.

Les *pertinences* désignent la façon dont est donnée la mesure (Boudon et al. 2000, p.119-123). Par exemple dimensionner une longueur de bassin en appliquant à la lettre une norme olympique est une *pertinence*.

Dans ce système entre *référence*, *pertinence* et *dimension*, la fonction dite d'*embrayage* est celle par laquelle un concepteur rend réalisable le projet. Cette fonction désigne le passage au réel de l'objet conçu (Boudon et al. 2000, p.98).



**Figure 55: Proposition de schématisation des concepts de l'architecturologie permettant de décrire un opération d'attribution de mesure, ici pour l'exemple de la conception de la largeur d'un bassin en référence aux normes olympiques**

Ces éléments interviennent dans l'activité d'attribution de mesures au travers d'opérations de conception dites élémentaires, qui sont:

- *Le découpage* (Boudon et al. 2000, p.114-118): opération par laquelle est choisi un point de vue pour penser son projet. Par le *découpage* sont décidées des *dimensions* sur lesquelles porteront les mesures. Par exemple, quand je travaille sur un projet et décide de commencer par réfléchir à une des façades principales, j'effectue une opération de découpage identifiant la *dimension* « façade principale » sur laquelle je souhaite faire porter mes prochaines opérations de conception;

- *La référenciation* : opération par laquelle sont mises en œuvre les références ;

- *Le dimensionnement* : opération menant à associer une *pertinence* à une *dimension*.



Pour concevoir le bassin d'une piscine, par exemple, différentes opérations de conception peuvent être sollicitées et formalisées grâce au vocabulaire proposé par l'architecturologie. Ainsi, la profondeur du bassin peut être dimensionnée pour que des enfants aient pied ou pour qu'ils puissent plonger. En termes architecturologiques, on dira alors que la profondeur du bassin est une *dimension* à laquelle on donne une mesure par rapport à une *pertinence* technique (la poussée d'Archimède) ou à la taille d'un enfant (l'« échelle humaine »). De même, la largeur de notre bassin pourra être *dimensionnée* par rapport au nombre de lignes d'eau nécessaire : la *dimension* est ici la largeur et la *pertinence* de cette mesure est fonctionnelle si le but de la mesure est de permettre à « n » nageurs de s'exercer sans se gêner. La longueur du bassin pourra quant à elle être *dimensionnée* de manière à construire une « piscine olympique » de 50 m : la *dimension* longueur est alors *dimensionnée* en fonction d'une *référence* qui est la norme olympique (cf. Figure 55). Attention, la mesure se distingue de la contrainte, ainsi si le bassin dont on a pris l'exemple est contraint selon une norme olympique non choisie, alors il n'est pas sujet à des opérations de mesure (Boudon 1991).

Tous ces concepts visent à décrire, de manière *a priori*, la complexité des opérations d'attribution de mesure, entendue comme opérations de conception. *Dimensions*, *pertinences* et *références* sont propres à chaque opération, comme la manière de les mettre en œuvre *via* le *découpage*, la *référenciation* ou le *dimensionnement*. Cependant, certains domaines de référence peuvent être récurrents d'une opération de conception à l'autre. Les travaux du LAREA montrent qu'un certain nombre de classes d'opérations de conception peut être identifié. Ces classes d'opérations sont distinguées par des domaines de référence, comme par exemple la géométrie, la culture, le symbolique, etc. (Boudon et al. 2000, p.166).

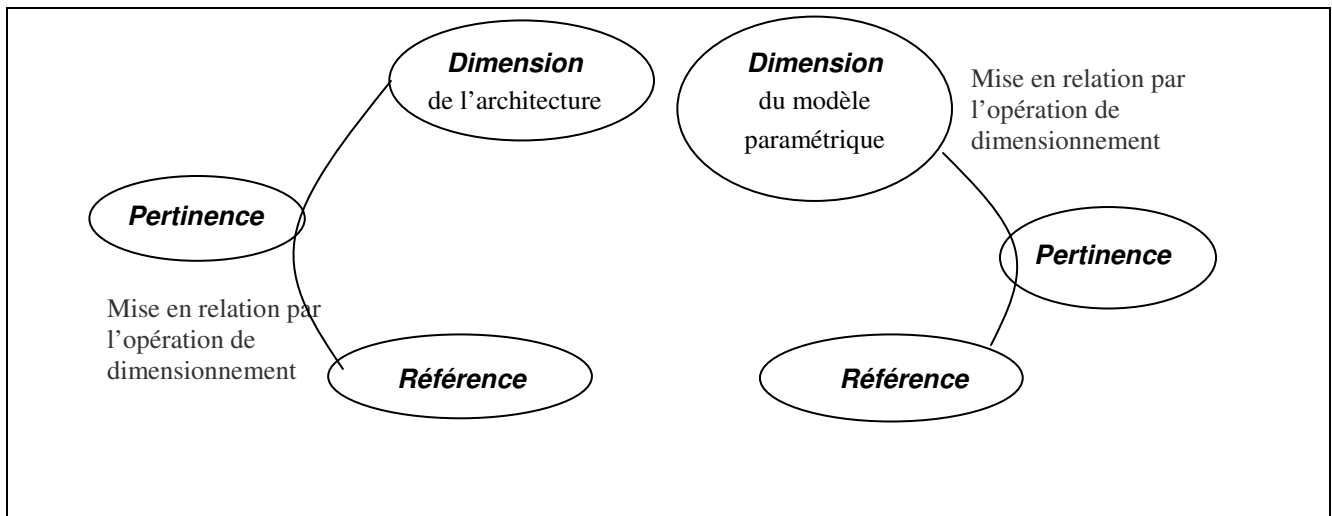
### 5.1.3 Conception architecturale et Conception de modèle paramétrique

Cette recherche interroge l'usage de la modélisation paramétrique en conception architecturale, en termes d'opérations cognitives de conception. L'appareillage théorique construit par l'architecturologie pour connaître la conception en termes d'*opérations d'attribution de mesures* est le support scientifique de ce travail. C'est à partir de ce support que sont positionnés les concepts : *-d'opérations de la conception architecturale*, repris à la littérature architecturologique (Boudon 2002; Boudon et al. 2000; Boudon et al. 2005), et *-d'opérations de conception de modèles paramétriques*, concept construit par le présent travail de recherche.

Par *opération de la conception architecturale*, sont entendues les opérations cognitives d'attribution de mesure à un objet ou un espace physique architectural (Boudon 2002; Boudon et al. 2000; Boudon et al. 2005). Par *opération de conception de modèles paramétriques* sont entendues les opérations cognitives d'attribution de mesures à un modèle informatique faisant appel à des mécanismes, dits paramétriques, basés sur la propagation (cf. Chapitre 2). Ces deux concepts sont des *opérations de la conception* (cf. 5.1.2) dont les *dimensions* (ce sur quoi porte la mesure) se distinguent : relatives à l'architecture pour l'un, relatives à un modèle paramétrique pour l'autre (cf. Figure 56).

Cette recherche s'appuie donc sur l'hypothèse qu'un modèle paramétrique puisse contenir de la mesure et soit le résultat, entre autres, de la mise en œuvre d'*échelles*, au sens architecturologique de classe d'opérations d'attribution de mesures. Par exemple, si l'on construit un modèle paramétrique

définissant une trame de cercles dont la valeur des rayons dépend de la proximité des cercles à un point de référence : on peut observer ici le *découpage* de la *dimension* « rayon des cercles » et, le *dimensionnement* de ces rayons par ce qui pourrait relever de l'*échelle* dite de *voisinage*<sup>57</sup>.



**Figure 56: Proposition de schématisation d'opérations de conception architecturale et de conception de modèles paramétriques**

Les concepts d'*opérations de la conception architecturale* et d'*opérations de conception de modèles paramétriques* sont utilisés dans la thèse en vue de répondre aux questions : « Quelles sont les opérations cognitives de conception impliquées lors de la conception de modèles paramétriques en conception architecturale ? Comment accompagner ces opérations et ces mécanismes chez l'architecte, en particulier par la pédagogie ? ».

Pour cela, les opérations de conception relatives à un objet architectural ainsi que les opérations de conception relatives à un modèle paramétrique sont recherchées dans les cas analysés. Ainsi le modèle d'analyse proposé est celui-ci : rechercher dans les cas analysés les émergences d'*opérations de conception architecturale* et d'*opérations de conception de modèles paramétriques*, en tentant d'identifier *dimensions*, *références* et *pertinences* impliquées, puis interroger les relations qui existent entre elles.

Les usages des aides existantes à la modélisation paramétrique pour la conception architecturale (les Samples et les Patterns, cf. Chapitre 4) sont également interrogés du point de vue des opérations cognitives de conception impliquées.

Les analyses des opérations de conception architecturale et des opérations de conception de modèles paramétriques ont été menées sur un corpus de cas d'usages de la modélisation paramétrique en conception architecturale. Ces cas sont issus de pratiques d'architectes et d'étudiants architectes.

Dans la suite du chapitre, nous présentons : la nature des données recherchées (5.1.4), la méthode de traitement des données (5.1.5), les différentes méthodes mises en œuvre pour recueillir ces données et

<sup>57</sup> L'*échelle de voisinage* implique : d'« attribuer des mesures par contiguité, relativement à des éléments appartenant au voisinage et induisant une continuité spatiale » (Boudon et al. 2000, p.171) .

les données composant finalement le corpus. Ces méthodes relèvent de : la mise en œuvre d'entretiens (5.2), d'observations participantes (5.3) et d'expérimentations (5.4).

## **5.1.4 Les données recherchées**

### **5.1.4.1 Nature des données recherchées : signes indiciels d'opérations de la conception**

Pour mener à bien l'analyse de ces cas et identifier les opérations cognitives de conception en jeu, nous mettons les concepts architecturologiques de la thèse à l'épreuve de cas observés. Cette mise à l'épreuve relève de « l'Architecturologie Appliquée », qui s'appuie sur les concepts de « l'Architecturologie Fondamentale » pour construire des méthodes de recherche (Lecourtois 2011).

La méthode que nous avons suivie consiste à lire des données recueillies pour les cas observés en recherchant des « traces » d'opérations de la conception. Ces « traces » sont pour nous des « signes indiciels » au sens de Peirce, c'est-à-dire des signes renvoyant à un objet absent (Everaert-Desmedt 1995, p.61). L'objet absent affecte le signe indiciel, comme par exemple la girouette est affectée par la direction du vent et en est l'indice (Everaert-Desmedt 1995, p.61) ou encore, l'empreinte d'un pied sur le sol peut être le signe du passage de quelqu'un. Cette méthode vise donc à explorer « l'espace vrai », phénoménologiquement perceptible (cf. Figure 54), à la recherche d'indices d'opérations de conception (Lecourtois 2008).

### **5.1.4.2 Type de corpus analysé**

Les signes indiciels que nous recherchons sont ceux d'opérations de conception architecturale et d'opérations de conception de modèles paramétriques.

Divers supports ont été mobilisés à cet effet :

#### **-Des descriptions orales de projets :**

Des discours descriptifs ont été recueillis. Ils sont issus de publications (discours de présentation de projets par des concepteurs) ou recueillis grâce aux entretiens menés dans ce travail de thèse. Ces entretiens ont été menés en vue de recueillir un discours descriptif du processus de conception d'un projet (« Nous avons commencé par... puis nous avons réalisé que ... ») et des modèles paramétriques développés (« Nous avons développé ce modèle pour... ici il y a... »). La méthode d'entretien est explicitée en 5.2.

#### **-Des modèles paramétriques de projets :**

Les modèles paramétriques et numériques ont été recueillis pour être analysés. Dans ces modèles paramétriques, les choix de méthodes de génération des géométries mais également l'organisation du modèle et les commentaires insérés dans ceux-ci sont autant de signes indiciels d'opérations de conception.

#### **- Des représentations graphiques de projet :**

Esquisses et images de rendu du projet ont également été utilisées dans ces analyses. Elles rendent compte d'opérations de conception ou de modélisation, par la schématisation d'un principe géométrique ou encore par l'annotation de captures d'écran du modèle numérique par exemple.

Ces données ont été recueillies selon trois méthodes : -l'entretien (5.2), -l'observation participante (5.3) et -l'expérimentation (5.4).

### 5.1.5 Méthode de traitement des données

La méthode de traitement des données mise en œuvre au cours de ce travail de recherche implique à la fois un travail d'objectivation de l'analyse (5.1.6) au travers de l'objectivation de la participation du chercheur et de celle des données recueillies ; ainsi qu'un travail d'analyse des données au travers de l'appareillage architecturologique développé (5.1.7). Les données issues des expérimentations font quant à elle l'objet d'analyses spécifiques, développées dans le paragraphe 5.4.3.

Comme illustration de ces méthodes de traitement, un exemple d'analyse de cas est donné à la fin de ce chapitre (5.5.1). Les analyses effectuées pour cette thèse sont intégrées dans l'annexe 2.

### 5.1.6 Pour une objectivation de l'analyse

La première étape du traitement des données est d'inscrire celui-ci dans une démarche d'objectivation.

Deux objectivations ont été effectuées : une objectivation de ma participation en tant que chercheur (« objectivation participante ») (Bourdieu 2003) et, une objectivation de la source des données.

Pierre Bourdieu définit l'*objectivation participante* comme « *objectivation du sujet d'objectivation, du sujet analysant, bref, du chercheur lui-même* » (Bourdieu 2003). Il s'agit ici d'objectiver le choix des données recueillies par le chercheur (Pourquoi telle information a retenu l'intérêt du chercheur plutôt qu'une autre ?) en fonction des conditions sociales du chercheur et de son positionnement dans le *champ* scientifique. Les choix scientifiques faits par le chercheur sont liés à sa position dans son champ professionnel (Bourdieu 2003). L'explicitation du champ scientifique dans lequel s'inscrit cette recherche et le contexte du choix des données sont donc autant d'éléments d'objectivation de notre analyse.

Cette objectivation participante est complétée par une objectivation des données issues des entretiens menés pour la recherche. De nombreuses recherches ont pointé les enjeux impliqués dans la production de discours (Foucault 1971). En particulier, dans le champ de la sociologie, Christophe Camus pointe les enjeux de discours d'acteurs de l'architecture visant à expliquer leurs activités (Camus 1996; Camus 1999). Pour lui, quand un acteur explique son activité, il se place dans une situation de représentation qui infère sur son discours (Camus 1999, p.107)<sup>58</sup>. La description du projet pouvant

---

<sup>58</sup> « Comme n'importe quel acteur social, un professionnel se présente et explique ce qu'il fait différemment selon ses interlocuteurs et la situation. » (Camus 1999, p.107).

même être un intermédiaire pour parler de soi (Camus 1999, p.107)<sup>59</sup>. Le discours sur son activité est pour l'acteur un lieu de représentation de soi, dépendant de son interlocuteur. Plusieurs éléments sont donc à prendre en compte pour pouvoir utiliser ces discours :

- les données peuvent être falsifiées par l'acteur dans une représentation de soi spécifique (pas par malice ou mensonge mais parce que le discours s'inscrit dans un contexte social) ;
- les données peuvent être falsifiées ou reconstruites à cause de leur déformation par la mémoire dans le cas de verbalisation *a posteriori* (comme c'est le cas des entretiens) ;
- les données dont rendent compte ces discours sont fragmentées et lacunaires.

Pour contrer ces différents biais, quelques stratégies peuvent être mises en place : - l'explicitation du contexte de production des entretiens pour l'analyse; et surtout - le contrôle des données par la vérification et le croisement de sources différentes.

Les discours d'acteurs dépendent du contexte de leur production. Le rôle du chercheur dans le contexte de production des données peut être objectivé par l'explication de ces contextes dans des comptes rendus d'entretiens (Blanchet & Gotman 2010, p.105).

Pour chacun des discours recueillis, nous avons donc rendu compte du contexte de leur construction. Par ailleurs, pour chacun des cas étudiés plusieurs sources ont été sollicitées en vue de recouper les informations recueillies.

### 5.1.7 Une méthode issue de l'Architecturologie Appliquée

Les données recueillies et objectivée (5.1.6), ont été analysées grâce à une méthode issue de l'Architecturologie Appliquée.

Cette méthode issue de l'Architecturologie Appliquée (Lecourtois 2011) vise à *interpréter*<sup>60</sup> la réalité empirique observée au moyen des concepts construits de manière *a priori*. Cette réalité empirique est abordée par la lecture de *signes indiciels* d'opérations de la conception (cf. 5.1.4.2).

Les concepts mis à l'épreuve de l'analyse de cas sont :

1- les concepts relatifs aux opérations de conception :

\* Les *opérations de la conception architecturale* (cf. 5.1.3) : qui sont les opérations cognitives d'attribution de mesures à un objet architectural (*découpage, dimensionnement, référencement*);

\* Les *opérations de la conception de modèles paramétriques* (cf. 5.1.3) : qui sont les opérations cognitives d'attribution de mesures à un modèle paramétrique;

\* Les relations entre les opérations : telles qu'elles sont définies dans la littérature architecturologique comme pouvant être des relations temporelles ou des relations spatiales (*surdétermination, juxtaposition, codétermination, relais, cascade*) (cf. 6.3) (Boudon et al. 2000, p.200);

---

<sup>59</sup> « certaines descriptions de bâtiments fonctionnent comme une représentation de soi » (Camus 1999, p.107).

<sup>60</sup> Dans son article Architecturological and epistemological research on collaborative design, Caroline Lecourtois définit l'Architecturologie Appliquée comme une « science interprétative ». Dans sa définition d'une science interprétative, Lecourtois décrit le rôle de l'interprétation: « An interpretative science gives tools to interpret realities through its scientific object that is to say, to build on them, focused knowledge. »(Lecourtois 2011).

2-les concepts relatifs aux spécificités de mise en œuvre de ces opérations de conception :

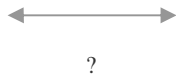
\* Les opérations logiques, en particulier les opérations de généralisation et de vérification ;

\* Les *opérations pragmatiques*, relatives à la conception de la situation de conception, telles que définies par Samia Ben Rajeb (Ben Rajeb 2012).

Dans leur ouvrage *Manuel de recherche en science sociale*, Quivy et Campenhoudt mettent l'accent sur la préparation des données pour leur analyse (Quivy & Campenhoudt 2006, p.193). Suivant, leur conseil, mon analyse se constitue :

- d'une description du projet et du modèle paramétrique développé visant à introduire le projet, son contexte et ses intentions ;

-d'une analyse des *opérations de conception* impliquées lors du processus : cette analyse est formalisée par un tableau (*cf.* Tableau) et distingue les opérations de la conception relatives à l'objet architectural ou au modèle paramétrique conçus. Cette analyse questionne également les relations et les structures de ces opérations.

| <i>Opérations de Conception Architecturale</i> | <i>Relation</i>  | <i>Opérations de modélisation paramétrique</i> |
|--|--|--|
| ...  |  | ...  |
| ...  | ...  | ...  |

**Tableau 17: Exemple d'analyse d'opérations cognitives de modélisation paramétrique et de conception architecturale, extrait de l'analyse du projet "Pavilion" (Studio P9)**

Outre les opérations de conception architecturale et les opérations de conception du modèle paramétrique, cette grille d'analyse permet d'interroger :

1- la réelle distinction entre ces opérations (conception architecturale et conception du modèle paramétrique peuvent-elles être des activités confondues ?),

2- les relations entre opérations de conception architecturale et opérations de conception du modèle paramétrique,

3- les opérations émergentes et ne pouvant être assimilées avec une opération de conception et, en particulier les opérations pouvant intervenir sur la conception dans ces situations d'usage de la modélisation paramétrique pour la conception architecturale, sans relever directement des deux types d'opérations de conception citées (comme par exemple des opérations logiques ou des opérations pragmatiques).

Les données disponibles étant relativement disparates d'un cas à l'autre, les analyses, si elles tiennent d'une même méthode, n'ont pas toutes mené au même degré d'approfondissement. Ainsi les résultats obtenus peuvent relever de la reconstitution lorsque les données recueillies sont très fournies, ou d'hypothèse de reconstitution quand les données recueillies sont peu nombreuses (Lecourtois 2008).

Dans un extrême comme dans l'autre, les analyses ne visent pas une reconstruction de la réalité, mais une approximation de celle-ci afin d'atteindre une compréhension et une caractérisation de la modélisation paramétrique pour la conception architecturale du point de vue des activités cognitives.

Cette méthode est utilisée pour traiter les données recueillies dans le cadre de cette recherche, celles-ci sont explicitées dans les sections suivantes.

## 5.2 Les entretiens

### 5.2.1 La méthode d'entretien

Des discours de description du processus de projet ont été recueillis grâce à des entretiens semi-directifs. La méthode de l'entretien semi-directif permet d'orienter l'interviewé sur des sujets précis, tout en lui laissant la liberté d'aborder des sujets ou des points non prévus. L'entretien semi-directif permet de mener une discussion dont les questions sont rédigées à l'avance et structurées par des « grilles d'entretien », tout en permettant à l'interviewé de répondre aux questions de façon ouverte et dans l'ordre qu'il souhaite (Quivy & Campenhoudt 2006; Blanchet & Gotman 2010).

Dans le cadre de cette recherche, nous avons mis en place des dispositifs pour permettre d'orienter l'entretien vers un « ordre du discours » (Foucault 1971) le plus adéquat possible : c'est-à-dire vers un discours narratif et descriptif, tentant d'éviter l'écueil des discours de normalisation, de justification ou de valorisation (Lecourtois, 2004). Ces dispositifs sont :

- le maintien d'une posture d'écoute attentive et curieuse, focalisée sur le travail (les projets, les modèles, les pratiques, *etc.*) plutôt que sur les personnes. Cela permet à l'interviewé de se concentrer sur la narration du processus de projet.

- le recours à des supports de discours concrets (des modèles informatiques, des représentations des projets abordés, *etc.*) qui permettent d'orienter le discours sur des descriptions de faits, plutôt que sur des représentations ou des théories plus ou moins normatives et fantasmées de ce qu'est ou doit être la pratique interrogée.

- le recours à des questions de relance très concrète (comme : « Qui a travaillé sur ce projet ? Pourquoi ? Quels sont leurs profils ? »).

Les grilles d'entretien ont été construites en fonction des questions de recherche et en vue de favoriser une narration du projet (*cf.* Tableau 18). Le recours à des questions très concrètes comme relances étaient fréquentes.



|  |  |  |
|--|--|--|
| Questions de la recherche :<br>Au sein de l'agence qui conçoit les modèles paramétriques ? Pourquoi ?  |  |  |
| Organisation générale de l'agence  | L'agence en générale                                   | <i>De combien de collaborateurs se constitue l'agence ? Quelles sont leurs profils ? (ingénieur, architecte, etc.)</i>   |
|  | Les acteurs de la modélisation paramétrique en général | <i>Qui construit les modèles paramétriques ? Pourquoi ? Quels sont leurs profils ?<br/>Si ce sont des acteurs différents de la conception, sur quelles phases du projet interviennent-ils ? Sur quels projets ?</i>  |
| Questions de la recherche :<br>Quelles sont les opérations de modélisation et les opérations de conception architecturale en jeu dans tel projet ? Quels sont les liens entre opérations de modélisation et opérations de conception pour tel projet ? |  |  |
| Choix d'un projet précis pour en discuter  | Choix du projet  | <i>Pouvons-nous maintenant parler d'un projet en particulier ? Aimeriez-vous parler d'un projet en particulier ? pourquoi ?</i>  |
|  | Missions   | <i>Description libre du projet (non-directif)</i>  |
|  | compétences  | <i>Qui a travaillé sur ce projet ? Quels sont leurs profils ? pourquoi eux ?</i>   |
|  | Processus  | <i>Comment s'est élaboré le travail de conception pour ce projet? Sur quels supports ? (esquisses papiers/maquettes physiques/prototypes/maquettes info....)<br/>Comment se constituent les premières hypothèses de conception ? (déterminations liées à la fabrication ? à la géométrie ? au site ? etc.)</i> |

**Tableau 18: Synthèse des questions posées lors des entretiens**

## 5.2.2 Population interviewée et données recueillies

### 5.2.2.1 Entretiens exploratoires et premiers entretiens sur les usages

Dès octobre 2009, suite à une première exploration de la littérature sur les usages de la modélisation paramétrique en architecture (Kolarevic 2005; Szalabaj 2000), l'hypothèse a été posée qu'en agences, la modélisation paramétrique était utilisée par des spécialistes et non par les architectes chargés de la conception. C'est à partir de cette hypothèse qu'ont été menés les premiers entretiens (cf. Tableau 19). A chaque entretien ou groupe d'entretiens du Tableau 19 correspond un code (entre crochets) qui renvoie dans les paragraphes suivants à un résumé du compte-rendu de l'entretien ainsi qu'à une rapide description des informations majeures recueillies.

|     |   |
|-----|---|
| [1] | 10 novembre 2009 _ Entretien avec Emilie Hergott (ingénieur-architecte, RFR) à propos de son travail sur le projet du Pavillon de la Fondation Louis Vuitton pour la Création.  |
| [2] | 13 novembre 2009_ Entretiens sur le site du chantier du Pavillon de la Fondation Louis Vuitton pour la Création (PLV), avec Emilie Hergott (ingénieur-architecte, RFR), Nicolas |

|     |   |
|-----|---|
|     | Leduc (ingénieur, TESS) et Lorenzo Ponzio (ingénieur, TESS), ainsi qu'avec Greighton Willis (architecte) membre de l'agence d'architecture locale : Studios Architecture. |
| [3] | 29 novembre 2009 _ lors de l'OpenWorkshop Smart Geometry, entretien avec Hugh Whitehead (fondateur et membre du Specialist Modelling Group, Foster and Partners)          |
| [4] | 29 novembre 2009 _ lors de l'OpenWorkshop Smart Geometry, Entretien avec Miriam Dalligna (architecte, Foster and Partners).   |
| [5] | 8 janvier 2010 _ Entretien avec Paul Ehret (architecte, Gehry Technologies Europ)   |
| [6] | 12 janvier 2010 _ Entretien avec Bernard Cache (architecte, Objectile)  |

**Tableau 19 : Entretiens semi-directifs avec des architectes utilisant la modélisation paramétrique dans le cadre de développement de projets d'architecture (novembre 2009-janvier 2010)**

**[1] Entretiens sur les pratiques liées au paramétrique dans le cadre du Pavillon pour la Fondation Louis Vuitton pour la Création (PLV)**

**\_ Entretien avec : Emilie Hergott (RFR)**

**Résumé du compte-rendu d'entretien :** Lors d'un premier entretien informel à l'ENSA de Paris la Villette, Emilie Hergott m'a présenté son travail au sein de l'agence RFR pour le projet Pavillon de la Fondation Louis Vuitton pour la Création (PLV). L'analyse de cet entretien m'a menée à construire d'autres questions, plus techniques et plus précises, soulevées lors d'un nouvel entretien (*cf.* Entretiens [2]).

**Informations majeures :** Organisation du travail dans l'équipe RFR sur le projet PLV, Organisation et description du projet PLV, introduction à Digital Project.

**[2] Entretiens sur les pratiques liées au paramétrique dans le cadre du PLV**

**\_ Entretiens avec : Emilie Hergott (RFR), Lorenzo Ponzio (TESS), Greighton Willis (Studios Architecture), Nicolas Leduc (RFR)**

**Résumé du compte-rendu d'entretien :** Ces entretiens avec différents membres des équipes travaillant sur le projet PLV ont été menés de façon relativement informelle. Après un premier entretien avec Emilie Hergott (*cf.* entretien [1]), un rendez-vous a été fixé sur son lieu de travail afin d'aborder plus précisément les modèles paramétriques sur lesquels elle travaillait. Suite à cette entrevue, un entretien a été mené avec Lorenzo Ponzio (RFR-TESS) car ce dernier avait plus d'expérience sur le projet. J'ai ensuite interrogé Greighton Willis au sujet de la relation entre l'évaluation structure et la conception architecturale. Greighton Willis est un des architectes de Studios Architecture, dont les missions sont le lot architecture et le lien avec Gehry Partners. Enfin, j'ai mené un entretien avec Nicolas Leduc (TESS) sur la conception de la structure des verrières (les « voiles »).

**Informations majeures :** Les entretiens, relativement informels, ont été fluides et riches en informations. Il s'agissait d'entretiens exploratoires visant à une acculturation au terrain, pour comprendre : l'organisation du travail au sein du projet PLV, l'utilisation de Digital Project pour le

travail en commun et le partage d'informations, les enjeux de ce partage de données, l'organisation du modèle Digital Project et son évolution au cours du projet.

### **[3] Entretiens sur les pratiques liées au paramétrique chez Foster and Partners et plus particulièrement au sein du Specialist Modelling Group**

#### **\_ Hugh Whitehead (Responsable du Specialist Modelling Group chez Foster and Partners)**

**Résumé du compte-rendu d'entretien :** Cet entretien a été mené dans le cadre d'une formation intensive à Generative Components, organisée par l'association Smart Geometry à la TU de Delft. Une grille de questions a été fournie à l'interviewé comme support de l'entretien. La retranscription de l'entretien est insérée en Annexe 1 p.327. La difficulté de la langue, le contexte exploratoire de l'entretien ainsi que le contexte académique de la rencontre ont participé à la mise en place d'une relation marquée étudiant-enseignant, mettant la discussion sur un plan normatif. Cependant, des informations décisives ont été communiquées du point de vue de l'établissement d'un aperçu général des pratiques de la modélisation paramétrique en architecture. De plus, la focalisation sur la description du projet City Hall (relativement ancien mais hors des enjeux de confidentialité) a été une bonne stratégie de construction d'un discours plus descriptif sur les pratiques du Specialist Modelling Group.

**Informations majeures :** Organisation de l'agence Foster and Partners et plus particulièrement de l'équipe Specialist Modelling Group. Description de l'usage de la modélisation paramétrique pour le projet City Hall ainsi que des relations entre modélisation paramétrique et conception architecturale.

La retranscription de l'entretien avec Hugh Whitehead est insérée en Annexe p. 327.

### **[4] Les pratiques liées au paramétrique chez Foster and Partners**

#### **\_ Entretien avec Miriam Dalligna (Architecte, Foster and Partners)**

**Résumé du compte-rendu d'entretien :** Cet entretien a eu lieu « au pied levé » lors d'une formation Generative Components à TU Delft (*cf.* entretien [3]), en réutilisant le guide conçu pour l'interview de Hugh Whitehead (*cf.* entretien [3]). L'entretien a été informel et le discours exploratoire quant aux pratiques de Foster and Partners.

**Informations majeures :** Pratiques de la modélisation paramétrique chez Foster + Partners, point de vue d'une architecte sur le Specialist Modelling Group dans l'agence, utilisation du numérique chez Foster and Partners.

La retranscription d'extraits de l'entretien avec Miriam Dalligna est insérée en Annexe.

### **[5] Les pratiques liées au paramétrique chez Gehry Technologies Europe (GTE)**

#### **\_ Paul Ehret (Architecte, GTE)**

**Résumé du compte-rendu d'entretien :** L'entretien a eu lieu au sein de l'agence sur le poste de l'interviewé et avec ses documents de travail comme support. Paul Ehret ayant l'habitude de communiquer sur les activités de Gehry Technologies Europ, l'entretien a été marqué par un discours préconçu et calibré pour valoriser le logiciel (Digital Project), les formations et les services de

consultance vendus par l'agence. Ce discours a été bénéfique pour l'analyse des missions de Gehry Technologies Europ.

**Informations majeures :** Pratiques de Gehry Technologies Europ, liens avec Gehry Partners, diversité des activités, description de la mission de Gehry Technologies sur le projet Atoll à Anger (rationalisation de la géométrie, aide à la fabrication)

## **[6] Les pratiques liées au paramétrique chez Objectile**

### **\_ Bernard Cache (Architecte, co-fondateur d'Objectile)**

**Résumé du compte-rendu d'entretien :** L'entretien a eu lieu au domicile de Bernard Cache. Le cadre informel n'a pas été propice à l'analyse de modèles ou de représentations graphiques. L'entretien a toutefois balayé assez largement les pratiques d'Objectile et ses différents projets (avec, en support, les productions exposées sur le lieu de l'entretien). Le discours a été normatif au début puis relativement libre.

**Informations majeures :** Intentions architecturales et philosophiques de Cache en général, diversité des usages de la modélisation paramétrique chez Objectile.

De ces entretiens résulte la confirmation de l'hypothèse de départ<sup>61</sup> : la modélisation paramétrique est, la plupart du temps, menée par des experts distincts des acteurs de la conception, pour les cas des agences Foster + Partners (*cf.* l'entretien de Hugh Whitehead [3]) et Gehry Partners (*cf.* l'entretien de Paul Ehret [5] et Greighton Willis [2]), ainsi que pour les moyennes et grandes agences comme Ateliers Jean Nouvel, AAVP et Oger International faisant appel aux services de Gehry Technologies pour « sous-traiter » leurs modèles paramétriques (*cf.* l'entretien de Paul Ehret [5]). La situation est différente pour le cas de l'agence Objectile, où la conception du modèle paramétrique est menée de front avec la conception architecturale (*cf.* l'entretien de Bernard Cache [6]), mais souvent sur des projets de moindre ampleur (installations artistiques, panneaux de bois, etc.).

---

<sup>61</sup> Une partie de ces résultats sur l'état des pratiques de la modélisation paramétrique en agence d'architecture ont été communiqués lors du séminaire des « journées ville » du PRES EST puis publié dans *Transformations des horizons urbains : savoirs, imaginaires, usages et conflits* (de Coninck & Deroubaix 2012). Une analyse des pratiques de la modélisation paramétrique pour le cas du projet du Pavillon Louis Vuitton a été publiée lors du colloque SCAN'10 (de Boissieu et al. 2010b).

### 5.2.2.2 Entretiens d'étudiants (Studio P9 et X-Over Studio)

Des travaux d'étudiants ont également été étudiés en vue d'analyser des processus cognitifs inscrits dans des contextes d'apprentissage de la modélisation paramétrique (cf. Tableau 20). Ces pratiques conjoignent modélisation paramétrique et conception architecturale.

|      |   |
|------|---|
| [7]  | 22 janvier 2010 _ Rendu final et présentation des projets des étudiants du X-Over Studio de l'universität die Angewandte de Vienne. |
| [8]  | 22 janvier 2010 _ Entretiens avec Philip Hornung et Siim Tuksam du X-Over Studio spécifiquement sur leurs projets.                  |
| [9]  | 14 décembre 2009 _ Entretien avec Marion Ott et Elodie Leroy du Studio P9 Digital Knowledge de l'ENSA Paris Malaquais               |
| [10] | 27 janvier 2010 _ Rendu final et présentation des projets des étudiants du Studio P9 Digital Knowledge de l'ENSA Paris Malaquais.   |

**Tableau 20:Entretiens semi-directifs et observations de rendus dans le cadre des Studio P9 et du X-Over Studio (décembre 2009-janvier 2010)**

#### **[7] Observation du rendu final du X-Over Studio (Fall semester 2009)**

**Résumé du recueil de données :** A l'occasion du rendu final du X-Over Studio, des discours sur les projets ont été recueillis. Ces discours sont de l'ordre de la justification et de la valorisation.

**Informations majeures :** Description des projets, des intentions architecturales. Données propres à décrire des opérations de conception architecturale.

#### **[8] Entretiens sur des cas précis de projets d'étudiants du X-Over Studio \_ Philip Hornung et Siim Tuksam (X-Over Studio, Fall semester 2009)**

**Résumé du compte-rendu d'entretien :** Suite à la présentation des projets du X-Over Studio, quelques projets intéressants du point de vue de l'usage de Digital Project ont été identifiés. Les deux entretiens ont été fluides et informels. Ils ont été menés au sein de l'école, dans l'atelier des étudiants et sur les postes respectifs des interviewés. Cela a permis de faire porter le discours sur la description des modèles paramétriques et des évolutions du projet. Ces entretiens ayant été menés après l'évaluation de leur rendu et l'intervieweur n'ayant pas pris part au jury, les enjeux de valorisation semblent avoir été majoritairement évacués.

**Informations majeures :** Description des projets, des intentions architecturales. Données propres à décrire des opérations des modèles paramétriques. Descriptions des évolutions du projet, de sa temporalité et des choix des logiciels (lequel à quel moment, pourquoi).

### **[9] Entretiens sur des cas précis de projets d'étudiants du Studio P9 Digital Knowledge \_Marion Ott et Elodie Leroy (Studio P9)**

**Résumé du compte-rendu d'entretien :** Les étudiantes travaillant en binôme, l'entretien a eu lieu avec les deux interviewées en même temps. L'entretien a eu lieu avant un des cours du Studio et a tout d'abord porté sur les modalités du cours, puis sur leur projet.

**Informations majeures :** Ces discours ont été intéressants quant à l'explicitation du contexte du Studio P9 (et en particulier sur les modalités de leur formation à DP, ses attendus, les intervenants, *etc.*). Des données de descriptions du projet développé par le binôme ont été recueillies (description orales, modèles paramétriques).

### **[10] Observation du rendu final du Studio P9 Digital Knowledge**

**Résumé du recueil de données :** A l'occasion du rendu final du Studio P9, des discours sur les projets ont été construits par les étudiants. Il s'agissait de discours de justification et de valorisation. Néanmoins ces discours étaient précis et descriptifs, ils ont pu être utilisés pour l'analyse de cas.

**Informations majeures :** Des discours propres à l'analyse d'opérations de conception architecturale et de conception du modèle paramétrique ont été recueillies pour les cas « Les particules en actions » de Constance de Batz et Hugo Houplain, « Concentricité et homothétie interne » de Thomas Perez et Nicolas Ruiz Gonzales et « Blooming Surface » de Maria-Thala Al-Aswad et Denice Martinez.

La population étudiée dans les cadres du Studio P9 et du CrossOver Studio est intéressante car il s'agit d'étudiants ayant suivi la même formation au modeleur paramétrique Digital Project (même équipe de formateurs, mêmes modalités d'enseignement). Les discours recueillis lors des rendus de ces studios ainsi que les entretiens recoupés avec d'autres données (modèles paramétriques et représentations graphiques), ont permis de mener à bien des analyses satisfaisantes des activités cognitives en jeu lors de l'usage de Digital Project pour ces projets d'architecture<sup>62</sup>.

### **5.2.2.3 Entretiens sur les usages et les opérations**

Des entretiens complémentaires et approfondis ont été menés (*cf.* Tableau 21), suite aux premiers entretiens exploratoires (*cf.* Tableau 19).

|      |   |
|------|---|
| [11] | 5 février 2011 _ Entretien avec Milovan Yanatchkov de l'agence RVBA sur le projet « Babel » |
|------|---|

---

<sup>62</sup> Les résultats de ces analyses ont été publiés dans deux articles : « Enseigner la conception architecturale avec la modélisation paramétrique : Quelle spécificité cognitive ? » publié dans les actes du colloque 01' design en mai 2010 (de Boissieu et al. 2010a) et « Cognitive operation of parametric modelling and/or cognitive operation of architectural conception ? » publié dans les actes du colloque eCAADe en septembre 2011 (de Boissieu et al. 2011).

|      |   |
|------|---|
| [12] | 7 juin 2011 _ Entretien avec Hugh Dutton (architecte, HDA) et Francesco Cingolani (architecte, HDA) sur l'agence en général et sur les pratiques liées au paramétrique en particulier.  |
| [13] | 7 juin 2011 _ Entretien avec Francesco Cingolani (architecte, HDA) plus précisément sur le processus de conception de la trame de façade du projet « Winter Garden ».   |
| [14] | 8 juin 2011 _ Observation d'une réunion de travail entre Francesco Cingolani (architecte, HDA), Phil Barrett (architecte, HDA), Pierluigi Bucci (ingénieur-architecte, HDA) et Hugh Dutton (architecte, HDA) pour la conception de la couverture du projet « Dazonghli ».   |
| [15] | 8 juin 2011 _ Interview de Pierluigi Bucci (ingénieur-architecte, HDA) sur ces pratiques de programmation sur les projets « Terna », « Winter garden » et « Orchard Road ».   |
| [16] | 13 juin 2011 _ Entretien avec Alexandre Pachiaudi (Architecte, Renzo Piano Workshop Building) sur les pratiques liées au paramétrique dans l'agence et en particulier sur le projet « Fondation Jérôme Célou Pathé ».   |
| [17] | 14 juin 2011 _ Entretien avec Gilles Desèvedavy (architecte) sur les pratiques liées au paramétrique chez Francis Solers de 1986 à 1993 et chez ReSie de 1993 à 2000. Entretien en particulier sur les projets « Mutations Amorphes » de ReSie, « TGBM quai Branly » de Solers et le projet pour l'école d'architecture de Venise de ReSie. |
| [18] | 23 juin 2011 _ Entretien avec Rick Smith (ingénieur, McNeel) qui a introduit CATIA chez Gehry Partners et l'a utilisé dans/pour l'agence de 1991 à 2001.  |
| [19] | 11 juillet 2011 _ Entretien avec Alexandre Pachiaudi (Architecte, Renzo Piano Building Workshop) et Gaétan Kohler (architecte, HDA) sur le projet « Abri n°177 » de leur collectif (OZcollectif).   |
| [20] | 21-24 septembre 2011 _ Entretiens avec Angelos Chronis (architecte), Anna Laskaran (architecte) et Sebastian Gmelin (architecte) à propos de leurs expériences dans le Specialist Modelling Group chez Foster and Partners. Entretiens informels, effectués lors du colloque eCAADe 2011 à Ljubljana.                                       |
| [21] | 15-05-2012 _ Entretien avec Angelos Chronis (architecte, Foster and Partners) et Francis Aish (ingénieur, Foster and Partners) de l'équipe Applied Research and Development chez Foster and Partners. Entretien formel au sein de l'agence.   |
| [22] | Juin 2012 _ Entretien avec Toshihiro Kubota (architecte, Ateliers Jean Nouvel)  |

**Tableau 21: Entretiens semi-directifs avec des architectes utilisant la modélisation paramétrique dans le cadre de développement de projets d'architecture**

### **[11] Entretien sur le projet « Babel » de RVBA**

**\_ Milovan Yanatchkov (architecte, RVBA)**

**Résumé du compte-rendu d'entretien :** L'entretien a été mené pour l'écriture d'un article pour la revue « Contraintes ». L'entretien a eu lieu dans l'atelier de l'interviewé, sur son poste avec comme support les archives du projet dont était visée la description.

**Informations majeures :** Des données propres à analyser les opérations cognitives de modélisation et de conception architecturale impliquées dans le projet « Babel » ont été recueillies. Cependant, les outils numériques utilisés (fluxus) n'ont pas été associés à des modeleurs paramétriques.

### **[12] Entretien sur les pratiques liées au paramétrique chez HDA**

**\_ Hugh Dutton et Francesco Cingolani (HDA)**

**Résumé du compte-rendu d'entretien :** L'entretien a eu lieu au sein de l'agence. L'agence HDA avait tenu à la présence de Francesco Cingolani, le « responsable communication » de l'agence sur le numérique. La discussion a été relativement fluide, portée par Cingolani mais librement ré-orientable par l'intervieweur. La discussion a, en particulier, été supportée par les images des projets affichées ou présentées dans un des books de l'agence.

**Informations majeures :** Pratiques et activités de l'agence, rôle de chacun des collaborateurs et compétences numériques de chacun. Informations générales sur la plupart des projets de l'agence.

### **[13] Entretien sur le projet « Winter Garden » de HDA**

**\_ Francesco Cingolani (HDA)**

**Résumé du compte-rendu d'entretien :** Suite à un premier entretien (*cf.* entretien [12]), une nouvelle discussion a été mise en place avec Cingolani sur une description précise d'un des projets de l'agence sur lequel l'usage d'outils numériques a été central: le projet Winter garden. Bien que l'interviewé n'ait pas eu ses fichiers avec lui, l'entretien a été précis sur les modèles développés. Des schémas ont parfois servis de support pour expliciter certains éléments de l'évolution du projet.

**Informations majeures :** Des données pertinentes pour l'analyse des opérations cognitives en jeu lors de la conception de la trame de toiture du projet Winter Garden ont été produites.

### **[14] Observation d'une réunion de travail sur le projet « Dazongli » de HDA**

**\_ Hugh Dutton, Phil Barrett, Francesco Cingolani, Pierluigi Bucci (HDA)**

**Résumé du recueil de données :** Suite aux entretiens menés auprès d'acteurs de l'agence HDA (*cf.* entretiens [12] et [13]), ceux-ci m'ont permis d'assister à une séance de travail portant sur le dessin de la couverture du projet Dazongli. La réunion s'est organisée en plusieurs temps : Barrett et Cingolani ont défini les questions à traiter lors de la réunion, puis Dutton est intervenu pour discuter des points. Des hypothèses de solutions ont été émises. Enfin, les questions de structure soulevées par les solutions proposées ont été discutées par Bucci. Il avait été convenu que les questions de l'intervieweur seraient posées à la fin pour ne pas perturber la réunion.



**Informations majeures :** Cette « observation passive » d'un processus de conception *in vivo* a permis de recueillir des informations propres à l'analyse des usages du numérique au sein de l'agence. Cependant, les outils numériques utilisés (RhinoScript) n'ont pas été associés à des modeleurs paramétriques.

#### **[15] Entretien d'un ingénieur sur son travail dans des projets précis de HDA**

**\_ Pierluigi Bucci (HDA)**

**Résumé du compte-rendu d'entretien :** Suite aux entretiens et observations menés au sein de l'agence HDA (*cf.* entretien [12], [13] et [14]), un entretien a été fixé avec Pierluigi Bucci pour traiter de la question des pratiques de la programmation dans l'agence. L'entretien a eu lieu sur le poste de travail de l'interviewé, où les programmes de celui-ci et les archives de l'agence ont servi de support à l'entretien. Après une présentation générale de différents travaux, des projets potentiellement intéressants du point de vue des questions de la recherche ont été repérés. Ces projets, en particulier « Terna », « Winter garden » et « Orchard Road », ont ensuite été approfondis.

**Informations majeures :** Les données recueillies sont propres à l'analyse des usages du paramétrique chez HDA et à l'analyse des opérations cognitives en jeu dans les projets « Winter garden » et « Orchard Road ».

#### **[16] Entretien sur les pratiques liées au paramétrique en général chez Renzo Piano Workshop Building (RPWB) et sur le projet Fondation Celu Pathé en particulier**

**\_ Alexandre Pachiaudi (architecte, RPWB, Collectif OZ)**

**Résumé du compte-rendu d'entretien :** L'entretien a porté sur les pratiques liées au paramétrique en général chez RPWB et sur le projet Fondation Celu Pathé en particulier. L'interviewé a eu recours à des schémas pour expliciter sa description du projet de la Fondation Jérôme Celu Pathé.

**Informations majeures :** Cet entretien a été intéressant pour l'analyse des usages du numérique chez RPWB et pour une analyse des usages de Digital Project et de la collaboration avec l'agence DesignToProduction pour le cas du projet de la Fondation Pathé.

#### **[17] Entretien sur les pratiques liées au paramétrique chez R&Sie**

**\_ Gilles Desèvedavy (architecte et enseignant, ENSA de Lyon)**

**Résumé du compte-rendu d'entretien :** L'entretien a eu lieu en extérieur, avec une grille d'entretien très générale. La discussion a porté sur le parcours et les pratiques de l'interviewé chez Francis Solers et R&Sie.

**Informations majeures :** les données recueillies sont propres à l'analyse des usages du numérique en général et du paramétrique en particulier chez Francis Solers de 1986 à 1991 et chez R&Sie de 1991 à 2001.

**[18] Entretien sur les pratiques de la modélisation paramétrique chez Gehry Partners de 1991 à 2001**

**\_ Rick Smith (anciennement Gehry Partners, aujourd'hui chez RhinoBIM)**

**Résumé du compte-rendu d'entretien :** Cet entretien a été improvisé suite à la communication de Rick Smith lors de la journée « Architecture et formes complexes » tenue à l'ENSA de Paris la Villette en juin 2012. La langue a présenté une difficulté mais les données visées étaient suffisamment spécifiques pour que l'entretien soit efficacement aiguillé vers celles-ci.

**Informations majeures :** Ce discours normatif a été intéressant pour l'analyse des usages du paramétrique chez Gehry Partners de 1991 à 2001.

**[19] Entretien sur les pratiques de la modélisation paramétrique du Collectif OZ**

**\_ Gaëtan Kohler (architecte, HDA, Collectif OZ), Alexandre Pachiaudi (architecte, RPWB, Collectif OZ)**

**Résumé du compte-rendu d'entretien :** L'entretien s'est déroulé de manière informelle auprès des deux acteurs simultanément. Les modalités de l'interview ont permis un recoupement direct des informations des deux interlocuteurs (débat, avis contraires sur les pratiques communes), ce qui a été intéressant.

**Informations majeures :** Cet entretien a été intéressant sur les limites de la pratique de la modélisation paramétrique éprouvées par ces architectes dans le cadre de leur pratique.

**[20] Entretiens sur les pratiques de la modélisation paramétrique chez Foster and Partners**

**\_ Angelos Chronis (architecte, Foster and Partners depuis 2010), Anna Laskaran (architecte, Foster and Partners pendant 1 an) et Sebastian Gmelin (architecte, Foster and Partners pendant 3 ans)**

**Résumé du compte-rendu d'entretien :** Lors du colloque eCAADe 2012 (Education and Research in Computer Aided Architectural Design in Europe) j'ai pu entrer en contact avec des architectes de chez Foster and Partners. Au cours de discussions informelles, des informations usuellement inaccessibles pour des raisons de confidentialité ont été recueillies. Néanmoins ces informations ne sont pas directement utilisables pour l'analyse : ni enregistrement ni prise de notes n'ont été permis par les interviewés.

**Informations majeures :** Ces entretiens informels ont été très intéressants pour l'analyse des usages de la modélisation paramétrique chez Foster and Partners, et en particulier pour l'analyse de l'évolution du « Specialist Modelling Group » (SMG) et sur les liens entre le SMG et les « Design Teams ».

**[21] Entretien avec Francis Aish et Angelos Chronis de l'équipe Applied Research and Development (ARD) chez Foster and Partners. Entretien formel au sein de l'agence.**

**Résumé du compte-rendu d'entretien :** L'entretien a eu lieu dans l'agence mais sans support spécifique. La grille d'entretien avait été communiquée à l'avance. L'entretien, négocié pendant plusieurs mois, a été formel.

**Informations majeures :** Les données recueillies ont été très intéressantes quant au fonctionnement de l'équipe ARD et de l'agence Foster and Partners et son rapport aux outils numériques.

La retranscription de l'entretien avec Angelos Chronis et Francis Aish est insérée en Annexe p.331.

## **[22] Entretien avec Toshihiro Kubota (architecte, Ateliers Jean Nouvel)**

**Résumé du compte-rendu d'entretien :** L'entretien a été mené au sein de l'ENSA de Paris la Villette, avec comme support de discussion un texte de mémoire produit par Kubota sur les pratiques informatiques des Ateliers Jean Nouvel. Le texte de mémoire a été produit dans le cadre de l'enseignement « Activité et Instrumentation de la Conception » dirigé par François Guéna et Caroline Lecourtois à l'ENSA de Paris la Villette en Master.

**Informations majeures :** L'entretien a permis, de façon informelle, d'éclairer certains points de son mémoire, en particulier sur l'utilisation de Digital Project et Grasshopper dans l'agence et sur le projet de marquise pour la Gare d'Austerlitz.

La population interviewée dans cette dernière série a été à même de communiquer sur des projets pertinents pour la recherche. Ces derniers entretiens sont riches en données propres à l'analyse des opérations cognitives impliquées par l'usage de la modélisation paramétrique dans des situations de conception architecturale.

## 5.3 Les observations participantes

### 5.3.1 La méthode de l'observation participante

Outre l'entretien semi-directif, des méthodes d'observation ont été sollicitées lors de cette recherche et, en particulier, une méthode d'observation participante. En science sociale l'« observation participante » décrit un mode de terrain nécessitant l'intégration active du chercheur dans la pratique ou dans le groupe social étudié. Ce mode de recueil de données est ambivalent. Il permet une grande proximité avec le terrain, et donc de recueillir de nombreuses données sinon inaccessibles. Il pose néanmoins la question de l'objectivation des observations du chercheur (Soulé 2007)<sup>63</sup>. Le chercheur est en effet en situation d'« être affecté » par le terrain d'étude ». La méthode utilisée pour pallier à cette ambivalence est celle du « compromis », visant à pondérer ce qui est investi dans la participation ou dans l'observation (Soulé 2007, p.137)<sup>64</sup>.

Notre terrain est très contraint par la confidentialité demandée de la part des maîtres d'ouvrage et des maîtres d'œuvre. De même, les discours des étudiants sur leurs travaux sont souvent marqués par une volonté de valorisation, visant à dissimuler les limites du travail. C'est pour cela que nous avons parfois eu recours à la méthode de l'observation participante.

Lors des quatre années sur lesquelles s'est déroulée cette recherche, nous avons été impliqués de façon active dans des pratiques de modélisation paramétrique pour la conception architecturale en milieu professionnel ainsi qu'en milieu pédagogique.

#### - En milieu professionnel :

Une observation participante a été menée pendant un mois à mi-temps dans l'agence DECODE. Nous avons été impliqués dans le développement d'un modèle paramétrique pour assister la phase exécution de l'ouvrage Canopée les Halles des architectes Berger et Anziutti. Dans cette situation, nous avons pu observer les enjeux de l'usage du modèle paramétrique Grasshopper dans un cadre professionnel, ceci impliquant entre autres des enjeux d'échange de données, d'archivage de versions, d'adaptation d'un modèle à de multiples entrées, d'adaptation du modèle paramétrique à des règles de géométrie spécifiques, etc.

#### -En milieu pédagogique, dans deux contextes :

1- Une observation participante de l'activité d'enseignement de la modélisation paramétrique a été menée dans le cadre de cours à l'ENSA de Paris la Villette. L'observation/participation de l'activité d'enseignement de la modélisation paramétrique a permis d'identifier des blocages, des difficultés et des points d'intérêt des étudiants ;

2- Dans le cadre d'Ateliers de projet faisant appel à la modélisation paramétrique à l'ENSA de Versailles (Atelier SP31), à l'ENSCI avec les projets « BioCity » et « observer » ainsi qu'à l'ENSA de

---

<sup>63</sup> « L'observation participante implique de la part du chercheur une immersion totale dans son terrain, pour tenter d'en saisir toutes les subtilités, au risque de manquer de recul et de perdre en objectivité » (Soulé 2007, p.128).

<sup>64</sup> « être participant et observateur à temps partiel, participant en public et observateur en privé. » (Soulé 2007, p.137).

Marne-la-Vallée (Cours Architecture paramétrique), nous avons pu intervenir auprès des étudiants comme « expert de la modélisation paramétrique » pour les accompagner dans la conception de modèles paramétriques pour leurs projets.

Les acceptions de l'observation participante sont multiples. Nous reprenons Soulé qui propose, pour décrire les différentes méthodes, de distinguer ce qui relève de la participation (quelle implication du chercheur ? quelle intimité avec le terrain ?), de l'observation (quelle prise de distance ?).

**- Degré de participation :**

La participation menée dans cette recherche a été, la plupart du temps, « couverte » (Soulé 2007, p.128): les architectes et les étudiants dont les pratiques étaient observées, s'ils connaissaient mon statut de chercheur et de potentiel observateur, ne connaissaient pas l'objet de ma recherche ni mes méthodes. Vis-à-vis des acteurs observés, j'ai été considérée comme un simple participant : enseignant ou architecte.

Ces participations ayant eu lieu après le début de mes recherches, je n'ai jamais été un « participant pur » de ces pratiques.

**- Degré d'observation :**

Dans toutes les situations d'observation participante menées, l'activité de participation était toujours effectuée à temps partiel : trois jours par semaine pour la pratique en agence et une demi journée par semaine pour les situations d'enseignement. L'observation, la prise de recul pour l'écriture et l'analyse ont toujours été menées en parallèle de l'observation.

Si les degrés d'observation et de participation semblent s'équilibrer, ces deux activités s'inscrivent néanmoins dans des temporalités différentes. Ainsi, dans les moments de participation, je me suis consacrée à mon activité de modélisation paramétrique et de collaboration avec l'architecte, même si je gardais en tête des hypothèses et des résultats. La prise de distance, quant à elle, se fait dans un temps différent : celui de la recherche, lors du retour au laboratoire, en cours et en fin de participation.

## **5.3.2 Données issues des observations participantes**

### **5.3.2.1 Observations participantes en agence**

**[O1] Observation participante d'un mois à mi-temps au sein de l'agence DECODE sur le projet Canopée les Halles.**

**Résumé du compte-rendu de l'observation participante :** Dans le cadre de la présente thèse, une observation participante d'un mois a été menée à mi-temps au sein de l'agence DECODE. Cette agence, comptant quatre salariés, est spécialisée dans le service relatif à la maquette numérique. Les collaborateurs développent des maquettes numériques en fonction des besoins et missions auxquelles l'agence répond. Lors de l'observation participante, j'ai été intégrée comme un salarié à mi-temps, dont la mission était d'intervenir sur le modèle Grasshopper développé par l'agence pour le projet Canopée les Halles. Ce modèle, développé dans un premier temps pour un bureau d'ingénierie, devait être contrôlé et mis à jour par rapport à l'avancée du projet.

**Informations majeures :** Les données principales recueillies sont relatives aux besoins et enjeux techniques de l'utilisation de Grasshopper pour un projet de grande envergure et aux intervenants variés.

### **5.3.2.2 Observations participantes en situation d'enseignement**

Au cours de la présente recherche, trois contextes d'enseignement ont été le lieu d'observations participantes :

- l'accompagnement de projets de diplômes de Master Spécialisé à l'ENSCI en 2011 et 2012,
- l'accompagnement de projets de 3eme année à l'ENSA de Versailles en 2012 et,
- l'accompagnement de projets de 4eme année lors du cours « Architecture Paramétrique » à l'ENSA de Marne la Vallée.

Ci-dessous sont présentés les projets observés lors de contextes pédagogiques pour lesquels j'ai également participé au développement des modèles paramétriques.

#### **[O2] Collaboration pour le projet « BioCity » d'Olivier Scheffer (ENSCI 2011)**

**Résumé du compte-rendu de l'observation participante :** Dans le cadre du diplôme de Master Spécialisé de l'ENSCI, une période d'observation participante a été menée dans le cadre d'une collaboration avec Olivier Scheffer pour son projet BioCity. Après un premier travail de conception d'Olivier Scheffer, je suis intervenue pour la phase de formalisation du projet *via* la modélisation paramétrique. Un travail de conception, à la fois du projet urbain et des modèles paramétriques, a été mené.

**Informations majeures :** Des informations sur la conception du projet ont été recueillies ainsi que des données relatives à la collaboration et au travail de « traduction » d'une intention.

#### **[O3] Accompagnement du projet « Observer » de Martial Marquet (ENSCI 2012)**

**Résumé du compte-rendu de l'observation participante :** Dans le cadre du diplôme de Master Spécialisé de l'ENSCI, une période d'observation participante a été menée dans le cadre de l'accompagnement du projet « Observer » de Martial Marquet. Il s'agissait ici d'accompagner Martial Marquet dans l'élaboration de sa stratégie de modélisation et de lui apporter un appui technique ponctuel.

**Informations majeures :** Cette observation participante a permis de tester certains supports d'enseignement et de hiérarchiser les connaissances nécessaires à la modélisation paramétrique.

**[O4] Accompagnement des projets « L'objet de la conversation » de Paola Assanti, Eunice Colella, Emilie Danel, Delphine Dargegen et Dani Jon, [O5] « Morphoclics » de Anne-Laure Farcy, John Carvalho, Mick Ben Susan, Julie Mistoco et Annael Médina, et [O6] « Danse-city »**

**de Marie Brossard, Alice Huet, Camille von Knechten, Elsa Favier, Marion Gomez et Caroline Michel (ENSA-v 2012) (ENSA-v 2012)**

**Résumé du compte-rendu de l'observation participante :** Dans le cadre du cours de projet SP31 de l'ENSA de Versailles, j'ai accompagné certains projets. Les étudiants n'ayant que des bases succinctes dans l'utilisation de Grasshopper, j'ai pu collaborer avec les différents groupes pour les aider à définir leurs stratégies de modélisation ainsi que pour leur fournir un appui technique et, au besoin, développer des éléments de modèles.

**Informations majeures :** Cette observation participante a permis de tester certains supports d'enseignement et de tester l'appropriation et l'évolution de modèles spécifiques dans le cadre de processus de projet.

**[O7] Accompagnement du projet « Topographies » d'Aurea Rodriguez, Pablo Gancedo, Samya Pelloquin et Mathias Saboya (ENSA Marne la Vallée 2012)**

**Résumé du compte-rendu de l'observation participante :** Dans le cadre de l'enseignement « Architecture Paramétrique » de l'ENSA de Marne-la-Vallée, je suis intervenue pour aider les étudiants à développer leurs modèles.

**Informations majeures :** Cette observation participante a permis de recueillir des données propres à l'analyse des opérations cognitives impliquées.

## **5.4 Les expérimentations**

Par rapport aux deux méthodes de recueil de données vues précédemment, l'entretien et l'observation participante, la méthode de mise en place d'expérimentation se distingue par ses objectifs. En effet, les expérimentations menées pour cette recherche visaient à recueillir des données propres à interroger l'utilisation de samples (des fragments de modèles, cf. Chapitre 4.1.2) pour concevoir des modèles paramétriques en conception architecturale.

L'utilisation des samples a été interrogée du point de vue de leurs pertinences pour l'apprentissage de la modélisation paramétrique, c'est pourquoi des expérimentations pédagogiques ont été mises en œuvre en niveaux Master 1 et 2 (cf. Tableau 22).

Trois expérimentations ont été menées dans le cadre d'enseignement de la modélisation paramétrique dans le champ « Science et Technique de l'Architecte » (STA) au sein de l'Ecole Nationale Supérieure d'Architecture de Paris la Villette (cf. Tableau 22).

| <i>Description de l'enseignement</i>   |           |                          |
|--|-----------|--------------------------|
| <b>Expe1</b>   |           |                          |
| ENSA Paris la Villette<br>Enseignant responsable : François Guéna<br>Cours de modélisation paramétrique « so823 », apprentissage de Grasshopper<br>*mars-juin 2011* étudiants de Master 1<br><br>// expérimentation focalisée sur l'usage de samples pour concevoir un modèle paramétrique, support d'un projet de conception d'une piscine                | 20h<br>TD | 4h d'expé<br>X 1 séance  |
| <b>Expe2</b>   |           |                          |
| ENSA Paris la Villette<br>Enseignant responsable : François Guéna<br>Cours de « scripting », apprentissage de Grasshopper et programmation de composants en VB<br>*sept- déc 2011* étudiants de Master 2<br><br>// expérimentation focalisée sur l'usage de samples pour concevoir un modèle paramétrique, support d'un projet de conception d'une piscine | 20h<br>TD | 4h d'expé<br>X 2 séances |
| <b>Expe3</b>   |           |                          |
| ENSA Paris la Villette<br>Enseignant responsable : François Guéna<br>Cours de modélisation paramétrique, apprentissage de Grasshopper<br>*mars-juin 2012* étudiants de Master 1  | 20h<br>TD | 4h d'expé<br>X 1 séance  |



|  |  |  |
|--|--|--|
| // Expérimentation focalisée sur l'appropriation d'un sample spécifique<br>inconnu des étudiants |  |  |
|--|--|--|

**Tableau 22 : Description des contextes pédagogiques d'inscription des expérimentations**

Pour les trois expérimentations, les étudiants, ont été formés à la modélisation paramétrique sur Grasshopper<sup>65</sup> (cf. 2.3.1 p.47) par François Guéna et moi-même pendant vingt heures de Travaux Dirigés (cf. Tableau 23). Une séance de travail a été consacrée à l'explication de la méthode des Patterns (cf. 4.1) et à l'utilisation des Samples. A la fin de l'enseignement, les étudiants ont été soumis à un test visant à évaluer leur capacité à utiliser Grasshopper en général, et les samples en particulier, dans un exercice de conception architecturale (cf. Figure 59). Ces tests sont pour nous des expérimentations.

---

<sup>65</sup> L'enseignement so921-2011 comptait en plus un enseignement du langage de programmation Visual Basic pour la programmation de nouveaux composants sur Grasshopper.

| Descriptif du cours so823 (mars-juin 2011) |   |
|--|---|
| Séance 1<br>110303                         | <b>Surfaces</b><br>Présentation de GH. Exemple du Gerkin<br>exercice : city hall  |
| Séance 2<br>110310                         | <b>Structure</b><br>Présentations des tableaux en programmation, division d'une surface selon deux paramètres (u et v)<br>Fin de l'exercice City Hall   |
| Séance 3<br>110317                         | <b>Morphing</b><br>Répétition d'un motif contenu dans une « boîte enveloppante » sur des « boîtes cibles » instanciées sur une surface<br>Exemple/Exercices du tissage cf NOX   |
| Séance 4<br>110324                         | <b>Programmation de fonctions : courbes</b><br>Création d'objets en 2D ou 3D à partir de fonctions<br>Génération aléatoire<br><i>Exercice</i> : musée de Doa : répéter des objets (disques circulaire) de façon aléatoire sur une courbe définie par une fonction<br>Utiliser une courbe à partir d'une fonction pour faire un « morphing » |
| Séance 5<br>110331                         | <b>Programmation de fonctions : surfaces</b><br>Corrigé de l'exercice sur le musée de Doa<br>(expérimentation sur les listes de points obtenus => comment modifier les listes pour les traiter différemment (ex tous les points ayant le même u ou le même v, etc))   |
| Séance 6<br>110407                         | <b>Attracteurs / paroi orientable</b><br>Contrôle d'un objet ou groupe d'objet grâce à sa position relative à un tiers  |
| Séance 7<br>110428                         | <b>Triangulation de Delaunay</b> et diagramme de Voronoi  |
| Séance 8<br>110505                         | <b>Evaluation et Optimisation</b> : Utilisation de Galapagos pour optimiser les surfaces des niveaux du Gherkin<br>Exercice pour optimiser certains des éléments structurels de la gare de Waterloo   |
| Séance 9<br>110512                         | <b>Evaluation et Optimisation (suite)</b><br>Correction de l'exercice sur Waterloo et travail sur une évaluation des courbures de surface   |
| Séance 10<br>110519                        | Pattern   |
| Séance 11<br>110526                        | Pattern   |
| Séance 12<br>110609                        | Examen : <b>Evaluation finale</b>   |
| Séance 12<br>110616                        | Examen : <b>Evaluation finale</b>   |

**Tableau 23 : Descriptif du cours type So 823**

L'enseignement de l'usage des Samples a été effectué avec le support d'un catalogue spécifiquement construit à partir des stratégies techniques enseignées pendant le cours. Ces notions techniques, « structurer un arbre de données », « répéter un motif par morphing », *etc.*, ont été formalisées sous la forme de vingt et un modèles paramétriques : les samples (*cf.* 4.1.2). Ces samples composent la bibliothèque du cours et ont été mis à jour après chaque expérimentation (*cf.* Figure 57), leurs évolutions ainsi que le catalogue finalement retenu sont décrits plus précisément dans le Chapitre 7, ainsi que dans l'annexe 3.

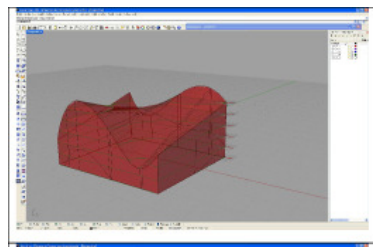
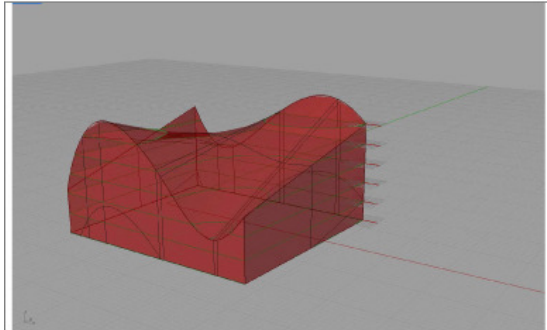
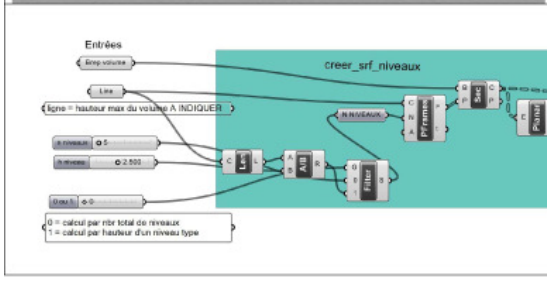
|  |  |  |   |                     |
|--|--|--|---|---------------------|
|   | <p><b>creer_srf_niveaux</b></p> <p>A partir d'un volume existant (BRep ou surface), créer des surfaces transversales à partir d'une verticale « maitre ». Ces niveaux peuvent être définis par deux méthodes : directement le nombre de niveau, ou la hauteur des niveaux.</p>                 | <p><b>Inputs :</b></p> <ul style="list-style-type: none"> <li>-Surface ou Brep</li> <li>-ligne « maitre » : verticale comprenant la hauteur maximale du volume</li> <li>-Entier variable : N sections</li> <li>-Réel variable : hauteur d'un niveau</li> </ul> <p><b>Outputs :</b></p> <ul style="list-style-type: none"> <li>- Surfaces</li> <li>- Courbes</li> </ul> | <p>Cours so823<br/>séance 8</p> <p>creer_srf_niveau<br/>x.ghx</p>   | <p><b>Expe1</b></p> |
|   | <p><b>creer_srf-sections_volume</b></p> <p>→ Crée des niveaux sur un volume existant (2 méthodes : par nbr de niveaux ou par hauteur d'un niveau type)</p> <p><b>Inputs :</b><br/>Volume, nbr ou h des niveaux, « arrêtes principale »</p> <p><b>Outputs :</b><br/>Srf</p> <p>Cf. SEANCE 9</p> | <p><b>GH v0.8.0006</b><br/><b>creer_srf-sections_volume.ghx</b></p>  | <p>Exemples :<br/>// Calculer la srf constructible d'un gabarit</p> | <p><b>Expe3</b></p> |
|  |  |  |   |                     |

Figure 57 : Samples « creer\_srf\_sections », présentations au sein du catalogue Expe1 (en haut) et Expe3 (en bas)

Les expérimentations ont eu lieu à l'ENSA de Paris la Villette, dans les salles habituelles du cours. Les Expe1 et Expe3 ont duré quatre heures chacune (cf. Tableau 22). L'Expe2 a duré huit heures divisées en deux sessions de quatre heures. Quatre heures étaient suffisantes pour que les étudiants montent une esquisse pouvant être considérée comme satisfaisante pour l'objectif de l'expérimentation. Le dispositif de deux fois quatre heures de l'Expe2 n'a, quant à lui, finalement pas été convaincant : les projets n'ont pas été plus aboutis.

Durant l'expérimentation, les étudiants étaient en binôme : l'un manipulait le modelleur et l'autre était chargé de prendre des notes manuscrites sur les choix de modélisation et de conception. Les binômes et la répartition des rôles étaient libres. Ce mode opératoire visait à amener les étudiants à verbaliser leurs intentions et décrire leurs actions.

Les étudiants étaient mis en situation d'examen. En début de séance leur ont été précisés : l'intitulé de l'exercice, les exigences de l'exercice et les ressources disponibles.

**-Intitulé de l'exercice**

Il s'agit de concevoir une piscine sur Grasshopper. Le projet doit tenir compte du site construit sur Rhinocéros et fourni sous la forme de fichier (cf. Figure 58). Seul(s) le(s) bassin(s) doivent être conçus ; accueil, billetterie et vestiaires étant intégrés dans les bâtiments déjà existants (cf. Figure 58). Le nord ainsi que l'indication d'une vue dégagée étaient intégrés dans le fichier du site.

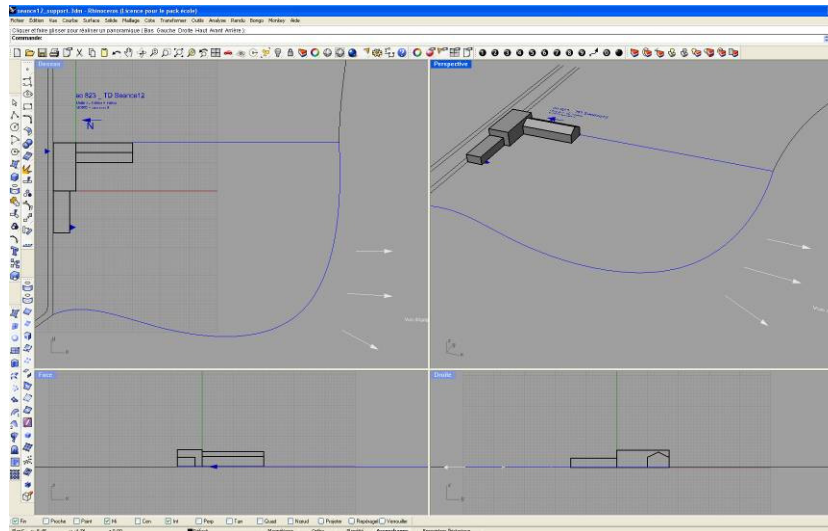


Figure 58 : Capture d'écran du fichier .3dm fourni comme support de l'exercice

### - Exigences de l'exercice

Les exigences de l'exercice étaient décrites aux étudiants en début de session par un exposé oral et une fiche récapitulative (cf. Figure 59). Il était demandé aux étudiants de répondre à l'exercice de conception en concevant un modèle sur Grasshopper. La modélisation devait solliciter au minimum trois samples issus de la bibliothèque du cours ou de ressources internet. Devaient être rendus en fin de séance : un fichier Grasshopper fonctionnant sans autres informations sur Rhinocéros que celles comprises dans le modèle du site fourni ainsi que, les notes prises durant l'exercice et le questionnaire fourni en début de séance (cf. Figure 60).

### - Ressources à disposition des étudiants

Les étudiants pouvaient, à tout moment, faire appel aux enseignants (François Guéna et Aurélie de Boissieu) pour répondre à leurs questions sur l'usage du logiciel. La bibliothèque de samples était à leur disposition sur le site de l'enseignement, ainsi que des liens url vers d'autres sites regroupant des modèles et des Patterns intéressants, présentés lors de la séance précédente.

**MODALITES DE L'EXERCICE :**

Durée : **3h** (de 9h30 à 12h30)

Sont à rendre:

- un (ou plusieurs) **fichier(s) Grasshopper**
- le **questionnaire** rempli
- les **notes** prises décrivant vos choix

Description :

Vous devez être par binôme :

- **l'un manipule** Grasshopper,
- **l'autre prend des notes** sous formes de points synthétiques, décrivant vos choix architecturaux et vos choix de modélisation.

Cette évaluation va être utilisée pour **une étude à l'ARIAM LAREA**. Pour cela:

- vos discussions seront enregistrées,
- des captures d'écrans de votre poste seront prises régulièrement (par CamStudio).

**DESCRIPTION DE L'EXERCICE :**

Vous devez concevoir une **piscine** :

- Le nombre de bassin est libre
- Elle peut être en plein air ou non
- Elle doit s'inscrire dans le **site** proposé dans le fichier rhino fourni
- L'accès, les sanitaires, les vestiaires etc. ne sont pas à concevoir : ils sont situés dans les bâtiments existants

Il vous est demandé :

- De **prendre en compte les spécificités du site**
- D'**utiliser au minimum 3 patterns différents**, venant indifféremment de la bibliothèque du cours ou d'autres sources (bibliothèques personnelles, sites web, etc.)

**Figure 59 : Intitulé de l'examen fourni aux étudiants lors des Expe1 et Expe2**

Le protocole de ces expérimentations visait à recueillir des traces d'opérations cognitives (de conception architecturale, de modélisation paramétrique ou encore de logique ou de collaboration), au travers :

- des notes manuscrites et schémas effectués durant l'expérimentation,
- des modèles paramétriques,
- des enregistrements audio des expérimentations et,
- des vidéos de captures d'écran effectuées par le logiciel CamStudio<sup>66</sup> tout au long de l'exercice.

<sup>66</sup> Logiciel libre faisant des captures d'écran systématiquement à une fréquence choisie. Ici la fréquence choisie a été d'une capture toutes les 13 secondes, ceci afin d'obtenir des vidéos de moindre poids. En effet CamStudio ne prévient pas en cas de bug lié à une surcharge de mémoire : on ne se rend compte du problème que lors de l'enregistrement de la vidéo, à la fin de la séance, les données sont alors perdues.

Des données sur les étudiants eux-mêmes ont également été recueillies à partir d'un questionnaire (cf. Figure 60). Ce questionnaire s'attachait à retracer le parcours de l'étudiant, par:

- L'identification de l'étudiant (nom, mail),

- La reconstitution de son parcours (Erasmus ? Précédent diplôme ? Travail en agence ?), ceci en particulier pour identifier les étudiants suivant des formations d'ingénieurs-architectes ainsi que pour évaluer les connaissances préalables de l'étudiant en modélisation et en programmation ;

- Provoquer des retours ouverts et plus personnels de la part de l'étudiant : quelques lignes étaient à leur disposition pour répondre aux questions suivantes : « Pourquoi avez-vous choisi cet enseignement ? Qu'en attendiez-vous ? » et « Pensez-vous utiliser Grasshopper pour le projet ? Pourquoi ? ».

**QUESTIONNAIRE**

Prénom.....  
Nom.....  
n° d'étudiant.....  
Adresse mail.....

si Erasmus, établissement d'origine :.....  
si précédent diplôme, préciser :.....  
si travail en agence, préciser :.....

Aviez-vous déjà utilisé Grasshopper ? Dans quelle mesure ? (seulement un aperçu ? logiciel utilisé en agence ? à l'école pour le projet?)  
.....  
.....

Savez-vous programmer ?.....

Quels logiciels utilisez-vous habituellement pour le projet ? .....  
.....  
.....

Pourquoi avez-vous choisi cet enseignement de Grasshopper ? Qu'en attendiez-vous ?  
.....  
.....  
.....

Pensez-vous utiliser Grasshopper pour le projet ? Pourquoi ?.....  
.....  
.....  
.....

Figure 60 : Questionnaire fourni aux étudiants lors de l'expérimentation

### 5.4.1.1 Réajustements des Expe 3 et 4 suite aux premières expérimentations

Le protocole de l'Expe3 a été adapté aux résultats obtenus lors des deux expérimentations précédentes. La question de l'appropriation d'un sample spécifique a été posée comme nouvelle problématique de l'analyse. Pour cela, l'intitulé de l'exercice a été transformé : il était demandé aux étudiants d'utiliser spécifiquement le sample « Panel-dispatch » issu de la plateforme [co-de-it]. Ce sample n'avait jamais été vu en cours. Cet exercice visait à la fois à évaluer les difficultés éprouvées par les étudiants pour comprendre et s'approprier un modèle qui n'était pas le leur, ainsi que pour évaluer comment chaque étudiant s'appropriait, ou non, le sample en fonction de son projet et de son modèle paramétrique.

Des ajustements dans le discours et les objectifs ont été également mis au point lors des sessions suivantes. En particulier, la nécessité de créer un fichier Grasshopper indépendant, c'est-à-dire sans modélisation sur Rhinocéros a été appuyé à l'oral.

De même, le mode de recueil de données a été affiné. L'expérimentation ayant lieu dans les salles de cours habituelles, le nombre d'ordinateurs disponibles était supérieur à celui des binômes. Lors de l'Expe2, plusieurs binômes se sont accaparés un poste « en plus ». Si un poste restait le poste « maître » sur lequel le modèle était conçu, le deuxième poste servait aux explorations sur internet en vue de l'appropriation des Samples. Cet usage spontané d'un poste spécifique pour rechercher des ressources pour la conception du modèle paramétrique s'est révélé intéressant, nous l'avons donc laissé faire et l'avons intégré au recueil de données lors de l'Expe3. Cela nous a permis d'analyser également les recherches de Patterns *via* les medias de diffusion (sites, google, bibliothèque du cours...).

### 5.4.2 Données recueillies lors des expérimentations

Lors de ces expérimentations, tous les questionnaires, les notes et les modèles paramétriques ont été recueillis. Cependant, tous les binômes n'ont pas été l'objet d'enregistrement audio et, si tous les écrans devaient être enregistrés grâce à CamStudio, suite à des problèmes techniques (poids des vidéos, bug du logiciel et parfois incompatibilité de raccourcis), toutes les vidéos n'ont pas été recueillies. Finalement, sur les vingt-sept binômes, seuls moins d'une dizaine présentent un corpus de données complet (*cf.* Tableau 24).

| <i>Groupes</i> | <i>Questionnaire</i> | <i>fiche</i> | <i>Vidéo</i> | <i>Audio</i> |
|----------------|----------------------|--------------|--------------|--------------|
| A1             | X                    | X            | -            | o            |
| B1             | X                    | X            | o            | x            |
| C1             | X                    | X            | x            | o            |
| D1             | X                    | X            | o            | x            |
| E1             | X                    | X            | o            | o            |
| F1             | X                    | X            | x            | o            |
| A2             | x                    | x            | x            | x            |
| B2             | X                    | X            | -            | x            |
| C2             | X                    | X            | x            | -            |
| D2             | X                    | X            | -            | o            |

|   |   |   |   |   |
|---|---|---|---|---|
| E2  | X | X | - | o |
| A3  | X | X | x | o |
| B3  | X | X | - | x |
| C3  | X | X | x | - |
| D3  | X | X | - | o |
| E3  | X | X | - | o |
| F3  | X | X | x | o |
| G3  | X | X | - | o |
| H3  | X | X | x | o |
| A4  | X | X | o | o |
| B4  | X | X | o | o |
| C4  | X | X | o | o |
| D4  | X | X | o | o |
| E4  | X | X | o | o |
| F4  | X | X | o | o |
| G4  | X | X | o | o |
| H4  | X | X | o | o |
| <p><b>Légende :</b></p> <p><b>O = manquant</b><br/> <b>- = incomplet</b><br/> <b>x = présent et utilisable</b></p> <p><b>A1 à F1 : Binômes EXPso823-2011 session 1</b><br/> <b>A2 à E2 : Binômes EXPso823-2011 session 2</b><br/> <b>A3 à D3 : Binômes EXPso921-2011</b><br/> <b>A4 à H4: Binômes EXPso823-2012</b></p> |   |   |   |   |

**Tableau 24: Données recueillies lors des expérimentations**

Le corpus recueilli à partir des expérimentations menées se constitue donc de données (enregistrement audio, enregistrement vidéos, traces graphiques) portant les traces des processus de projet de vingt-sept binômes d'étudiants. Ce sont ces diverses traces, en particulier les enregistrements par captures d'écran du modèle paramétrique et de son évolution, qui sont le support de nos analyses des opérations de conception du modèle paramétrique et des opérations de conception architecturale. Ces analyses sont insérées dans l'Annexe 2 et leurs résultats sont décrits dans le Chapitre 7.

### **5.4.3 Spécificités du traitement des données des expérimentations**

Trois analyses des données ont été mises en place :

- une analyse des vidéos évalue la fréquence et la nature de recours à des aides extérieures (cours, internet, etc.) ;

- une analyse des modèles paramétriques produits interroge l'appropriation des Samples utilisés (quelles transformations ? quelles insertions dans le modèle global ? etc.),



- enfin, une analyse architecturologique, identifie et interprète des traces d'opérations de la conception du modèle paramétrique ou de l'architecture.

L'exploration de bibliothèques en ligne ou de fichiers archivés par l'étudiant a été étudiée au travers de l'analyse des vidéos de captures d'écran. Cette analyse repose sur un codage systématique de l'occurrence:

- d'ouverture de sites web relatifs à des Samples (site du cours ou références données en cours),
- de téléchargement et d'ouverture de fichiers de Samples et,
- de recherche d'autres références sur internet.

Le codage est un relevé d'actions décrites succinctement sur une ligne de temps au travers d'un tableau (*cf.* exemple de codage du cas A3 dans le Tableau 25). Le codage est soumis à certaines limites liées aux données disponibles : le nombre réduit de captures d'écran par minute ne permet pas de tout percevoir.

| <i>tps</i>   | <i>Recherche sur le site so823</i> | <i>Ouverture de fichiers</i>  | <i>Recherche d'autres références (internet)</i>  |
|--|------------------------------------|---|--|
| <b>Jour 1</b> (9h36-11h58) Modélisation de la couverture, du bassin. Début d'exploration pour « terrasse » |                                    |   |  |
| 10h29  | Téléchargement d'anciens exercices |   |  |
| 10h32  |                                    | Ouverture du fichier perot13oct.ghx (exercice paroi salle d'expo) et pero17nov.ghx<br>Copier coller dans fichier de travail |  |
| 10h42  |                                    | Récupération d'un fichier d'exercice précédent (salle d'exposition) comprenant un pattern co-de-it détourné                 |  |
| 11h13  |                                    | Ouverture du fichier midpoint.ghx (dans explorer clef usb)  |  |
| <b>Jour 2</b> (09h10 – 12h00) modélisation des terrasses-cabines de plage                                  |                                    |   |  |
| 09h20  |                                    |   | Design patterns GH (Andrew)  |
| 09h31  |                                    |   | Co de it, téléchargement de Panels dispatch_Co-de-iT   |
| 09h36  |                                    | Ouverture de dispatch_Co-de-iT  |  |
| 09h39  |                                    |   | [ParametricModel.com]  |
| 10h18  |                                    |   | [ParametricModel.com]<br>téléchargement de « compare every item in a list » (ouverture et exploration) |
| 10h25  |                                    | Copier coller de dispatch_Co-de-iT dans le modèle du TD<br>Explorations, transformations pour appropriation                 |  |
| 10h50  |                                    | Retour/observation du fichier dispatch_Co-de-iT original  |  |
| 11h12  |                                    |   | [ParametricModel.com]  |

**Tableau 25 : Extrait du codage des vidéos recueillies pour le groupe A3 de l'EXPso921-2011**

L'analyse des modèles paramétriques interroge l'appropriation de samples pour des cas particuliers de conception architecturale. Un premier relevé des samples utilisés par les binômes permet de rendre compte de la fréquence des utilisations (quels samples ont eu le plus de succès ? lesquels n'ont pas été utilisés ?) ainsi que de leur combinaison. Suite à ce premier codage, des cas particuliers d'appropriations de samples sont identifiés pour une analyse plus approfondie. Cette analyse porte sur ce qui a été supprimé, modifié et conservé des Samples d'origine dans les modèles produits.

Suite à ces premières analyses et aux observations faites durant les cours et le test, des extraits de processus de conception architecturale effectués par les étudiants ont été choisis pour être l'objet d'analyses architecturologiques détaillées. Ces analyses portent à la fois sur les conversations enregistrées, les schémas rendus, les modèles paramétriques produits et les vidéos de captures d'écran. Ces analyses visent à reconstituer des opérations cognitives de conception architecturale et de modélisation paramétrique. Ces « reconstitutions » sont des interprétations émises par le chercheur après un travail d'investigation (cf. 5.1.3).

## 5.5 Synthèse du corpus

Les trois méthodes de recueil de données mises en œuvre (entretien, observation participante et expérimentation) ont permis de recueillir des données variées (en termes de nature et de sources) sur des cas de conception de modèles paramétriques en conception architecturale.

Comme nous avons pu le voir, ces cas sont issus de contextes de pratiques pédagogiques et de contextes professionnels. La fin de ce chapitre propose de décrire, par cas, les données finalement rassemblées.

### - Synthèse du corpus de travaux d'étudiants

La modélisation paramétrique est utilisée de façon diverse et active en école d'architecture. Si la diffusion de ces modeleurs ne concerne pas tous les étudiants, certains cas d'usages observés dans des situations pédagogiques se sont révélés riches de diversité en termes d'opérations de conception architecturale comme en termes de modélisation paramétrique. Ces étudiants en architecture, pour la plupart inscrits en Master 1 ou 2, sont considérés ici comme ayant une expertise en architecture.

Les cas de travaux d'étudiants recueillis dans le corpus s'inscrivent dans des contextes pédagogiques divers qui se sont déroulés entre 2010 et 2012. Ils sont décrits chacun précisément dans l'annexe 2 de cette thèse. Ces cas font appel à l'usage de Digital Project ou Grasshopper. Les projets analysés sont synthétisés dans le tableau ci-dessous en fonction de leur contexte pédagogique:

| <i>Modeleur param. utilisé</i>                                    | <i>Nom du projet et des étudiants</i>  | <i>Données disponibles</i>                     | <i>Remarques sur le contexte pédagogique</i>   | <i>Renvoi à l'analyse détaillée</i> |
|---|--|--|--|-------------------------------------|
| <b>Studio P9, ENSA Paris Malaquais, 2010</b>                      |  |  |  |                                     |
| DP  | « Concentricité et homothétie interne » de Thomas Perez et Nicolas Ruiz Gonzales | - discours des étudiants                       | -enseignants : Ph. Morel et Ch. Girard<br>-objectif pédagogique : compréhension et maîtrise de la FAO  | Annexe p. 340                       |
|   | « Blooming Surface » de Maria-Thala Al-Aswad et Denice Martinez                  | - représentations produites pour le jury final | → l'apprentissage de DP n'est pas la finalité de l'enseignement mais un support de réflexion<br>-rendu de projet néanmoins demandé <i>via</i> une démonstration du modèle paramétrique | Annexe p.343                        |
|   | « Les particules en action » de Constance de Batz et Hugo Houplain               |  | → formation des étudiants et des enseignants à DP par Gehry Technologies Europ   | Annexe p.348                        |
| <b>Studio CrossOver, Universität die Angewandte, Vienne, 2010</b> |  |  |  |                                     |
|   | « Radical Reproduction »   | - Discours de rendu                            |  | Annexe p.353                        |

|  |   |   |   |               |
|--|---|---|---|---------------|
| DP   | de Daniela Kroenhert et Adam Orlinsky                 | <ul style="list-style-type: none"> <li>- représentations de rendu</li> <li>- entretien <i>via</i> mails</li> <li>-captures d'écran du modèle paramétrique</li> </ul>                                    | <ul style="list-style-type: none"> <li>-Atelier « transversal » → tous niveaux d'étudiants confondus</li> <li>- objectif pédagogique : concevoir une architecture « innovante »</li> </ul>              |               |
| DP Rhino   | « Pavilion » de Philip Hornung                        | <ul style="list-style-type: none"> <li>- Modèle paramétrique</li> <li>- entretien</li> <li>-discours de rendu</li> <li>- représentations de rendu</li> <li>- entretien <i>via</i> mails</li> </ul>      | <ul style="list-style-type: none"> <li>-rendu du projet demandé <i>via</i> des planches 2D imprimées</li> <li>→ formation des étudiants et des enseignants à DP par Gehry Technologies Europ</li> </ul> | Annexe p.360  |
| DP   | « Fold » de Siim Tuksam                               | <ul style="list-style-type: none"> <li>- Modèle paramétrique</li> <li>- entretien</li> <li>-discours de rendu</li> <li>- représentations de rendu</li> <li>- entretien <i>via</i> mails</li> </ul>      |   | Annexe p.363  |
| <b>Cours So823, ENSA Paris la Villette, 2011 (noté so823-2011)</b> |   |   |   |               |
| GH   | Sans titre<br>Rebecca Paulin et Sebastiaan Leenknecht | <ul style="list-style-type: none"> <li>- Captures d'écran</li> <li>- enregistrement audio</li> <li>- esquisses</li> <li>- texte d'intention</li> <li>- modèle paramétrique</li> </ul>                   | <ul style="list-style-type: none"> <li>-Enseignant : F. Guéna et A. de Boissieu</li> </ul>  | Annexe p.     |
|  | Sans titre<br>Toshi et Pierre                         | <ul style="list-style-type: none"> <li>- Captures d'écran</li> <li>- enregistrement audio</li> <li>- esquisses</li> <li>- texte d'intention</li> <li>- modèle paramétrique</li> </ul>                   | <ul style="list-style-type: none"> <li>-un cours d'informatique : l'objectif pédagogique était l'apprentissage du logiciel GH</li> </ul>  | Annexe p.389  |
|  | Sans titre<br>Victoire et Aymeric                     | <ul style="list-style-type: none"> <li>- esquisses</li> <li>- texte d'intention</li> <li>- modèle paramétrique</li> </ul>   |   | Annexe p. 393 |
| <b>Mastère spécialisé, ENSCI (2011 et 2012)</b>                    |   |   |   |               |
| GH   | « BioCity » d'Olivier Scheffer (2011)                 | <ul style="list-style-type: none"> <li>- Modèles paramétriques (final et intermédiaires)</li> <li>- représentations et discours produits pour le rendu</li> <li>//Observation participante//</li> </ul> | <ul style="list-style-type: none"> <li>-Projets très personnels aux étudiants</li> </ul>  | Annexe p.367  |

|   |   |   |   |               |
|---|---|---|---|---------------|
| GH  | « Observer » de Martial Marquet (2012)  | - Modèles paramétriques<br>//Observation participante//   | -intervention du chercheur comme « expert en modélisation paramétrique » → importance du travail de collaboration   |               |
| <b>Studio SP31, ENSA Versailles, 2012</b>                               |   |   |   |               |
| GH,   | « L'objet de la conversation » de Paola Assanti, Eunice Colella, Emilie Danel, Delphine Dargegen et Dani Jon    | - Modèles paramétriques (final et intermédiaires)<br><br>- représentations et discours produits pour le rendu<br><br>//Observation participante// | - Enseignant responsable : C. Petetin   | Annexe p. 369 |
| GH  | « Morphoclics » de Anne-Laure Farcy, John Carvalho, Mick Ben Susan, Julie Mistoco et Annael Médina              |   | - cours de projet intensif (un mois)<br><br>-objectif pédagogique : initiation à la culture numérique et à des processus de conception croisant captation, FAO et programmation | Annexe p. 372 |
| GH  | « Danse-city » de Marie Brossard, Alice Huet, Camille von KNECTEN, Elsa Favier, Marion Gomez et Caroline Michel |   | - association de Grasshopper à des technologies processing et arduino   | Annexe p. 373 |
| <b>Cours « Architecture Paramétrique », ENSA Marnes la Vallée, 2012</b> |   |   |   |               |
| GH  | « Topographies » d'Aurea Rodriguez, Pablo Gancedo, Samya Pelloquin et Mathias Saboya                            | - Modèles paramétriques (final et intermédiaires)<br><br>- représentations et discours produits pour le rendu<br><br>//Observation participante// |   | Annexe p. 374 |

**Tableau 26: synthèse des cas de projets d'étudiants analysés du point de vue de l'architecturologie**

Les étudiants des ateliers « CrossOver » de l'Universitat die Angewandte (Vienne) et « P9 » de l'ENSA de Paris Malaquais ont la particularité d'avoir été formés à Digital Project par l'équipe de Gehry Technologies Europ, dans des objectifs pédagogiques distincts : concevoir un pavillon « innovant » pour le premier et, concevoir une installation construite pour le rendu pour le deuxième.

Les données issues des cas des projets du Master Spécialisé à l'ENSCI, du groupe « architecture paramétrique » de l'ENSA de Marne la Vallée et du « SP31 » de l'ENSA de Versailles ont la particularité d'avoir été recueillies par les méthodes de l'observation participante (cf. paragraphe 5.3).

#### **- Synthèse du corpus de travaux d'agences**

Les cas de conception en agence ont été retenus en fonction de la pertinence des appropriations de la modélisation paramétrique et des données recueillies. Il s'agit de cas :

-d'usage de la modélisation paramétrique en interne pour la conception : c'est le cas de la conception de la marquise de la gare d'Austerlitz par les Ateliers Jean Nouvel et de la conception du City Hall de Londres par Foster and Partners.

-d'usage de la modélisation paramétrique par des équipes externes intégrées à l'équipe de conception: c'est le cas de l'usage de Digital Project pour le projet du Qatar National Museum des Ateliers Jean Nouvel, effectué en collaboration avec Gehry Technologies.

-d'usage de la modélisation paramétrique par des experts pour des tâches spécifiques : c'est le cas de l'usage de Digital Project pour la conception des aménagements intérieurs et de la structure par les équipes spécifiquement choisies pour intervenir sur le Pavillon de la Fondation Louis Vuitton pour la Création, conçu par GP.

| <i>Lieu, date<br/>Intitulé du<br/>projet et<br/>concepteur</i>   | <i>Modeleur<br/>param.<br/>utilisé</i> | <i>Acteur de la<br/>modélisation<br/>paramétrique</i>              | <i>Données disponibles</i>   | <i>remarques</i>   | <i>Renvoi à<br/>l'analyse<br/>détaillée</i> |
|--|--|--|--|--|---|
| Paris,<br>Pavillon de la<br>Fondation<br>Louis Vuitton<br>pour la<br>Création<br><br>(Gehry<br>Partners)           | DP                                     | Concepteurs<br>assistés de<br>l'équipe de<br>Gehry<br>Technologies | - Entretiens avec plusieurs<br>acteurs (ingénieurs et<br>architectes chargés du projet<br>chez RFR, TESS et Studios)<br><br>- Arbre des spécifications du<br>modèle paramétrique   | -utilisation de la<br>modélisation paramétrique<br>dans un contexte particulier<br>de projet avec beaucoup<br>d'ambition et beaucoup de<br>moyen<br><br>- enjeux spécifiques sur le<br>partage de données et la<br>fabrication | Annexe p. 377<br><br>3.2.2                  |
| Paris, 2010-<br>2011 (non<br>construit)<br><br>marquise de la<br>gare<br>d'Austerlitz<br>(Ateliers Jean<br>Nouvel) | GH                                     | Cellule de<br>conception<br>numérique de<br>AJN                    | - Entretien de Toshi Kubota<br>(responsable de la cellule<br>« conception numérique »<br>d'AJN)<br><br>-travail réflexif (mémoire de<br>master) de T. Kubota<br>(Kubota 2012)<br><br>- Images issues du modèle<br>paramétrique | Usage de la MP pour des<br>tâches d'évaluation et de<br>rationalisation de la<br>géométrie   | Annexe p. 385                               |
| Doha, 2008-<br>2013<br><br>Qatar<br>National<br>Museum<br>(Ateliers Jean<br>Nouvel)                                | DP                                     | Equipe mixte<br>AJN et GT  | -dossier descriptif du modèle<br>(Kubota 2012)<br><br>- Communication<br>d'Edmondo Ochipinti (GT)<br><br>-entretien avec Paul Ehret<br>(GT)<br><br>-Entretien avec Toshi Kubota<br>(AJN)                                       | Usage de la modélisation<br>paramétrique pour le partage<br>d'informations, la<br>représentation, la<br>rationalisation et l'assistance<br>à la fabrication  | 3.2.3                                       |
| Londres,<br>1998-2002<br><br>City Hall   | DDD<br>(Micro<br>Station)              | SMG  | -Article de Hugh Whitehead<br>in (Kolarevic 2005)<br><br>-Entretien avec Hugh  | - distinction des acteurs de<br>la modélisation<br>paramétrique et de la<br>conception architecturale  | Annexe p. 386                               |

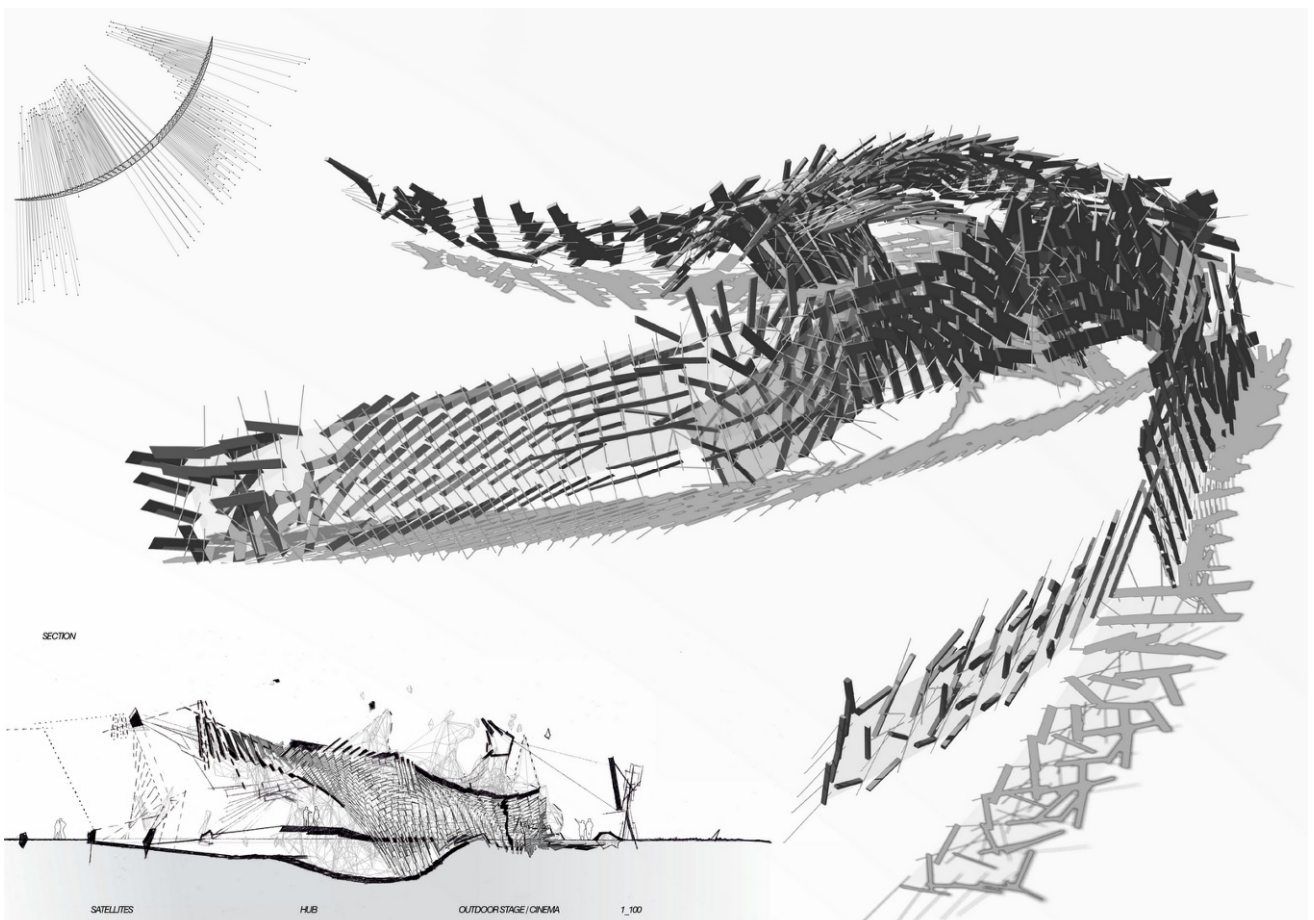
|                       |  |  |                              |   |  |
|-----------------------|--|--|------------------------------|---|--|
| (Foster and Partners) |  |  | Whitehead (de Boissieu 2009) | mais travail en commun de ces deux activités (par la collaboration) |  |
|-----------------------|--|--|------------------------------|---|--|

**Tableau 27: synthèse du corpus de cas d'usage de la modélisation paramétrique pour la conception architecturale en agence d'architecture**

Pour conclure ce chapitre, un exemple d'analyse architecturologique de cas d'usage de la modélisation paramétrique en conception architecturale est présenté ci-dessous.

### 5.5.1 Exemple d'analyse architecturologique : le cas « Radical Production » de Daniela Kroenhert et Adam Orlinsky (Universität Die Angewandte de Vienne)

Le projet « Radical Production » (cf. Figure 61) a été développé en 2010 dans le cadre d'un semestre du CrossOver Studio de l'Universität Die Angewandte de Vienne. Le projet répond à un programme de « pavillon innovant » implanté sur le site du parc Sigmund Freud à Vienne. Le modèleur paramétrique Digital Project a été utilisé pour ce projet.



**Figure 61: perspective et coupe du projet "Radical Reproduction", sources: Daniela Kroenhert et Adam Orlinsky**

Le processus de la conception, selon les étudiants du projet « Radical Production », a été organisé en plusieurs phases : 1- la conception de systèmes de calepinages (cf. Figure 62a), 2- la conception de surfaces constituant les espaces du pavillon (cf. Figure 62b), et 3- l'instanciation des calepinages sur les surfaces (cf. Figure 62c).

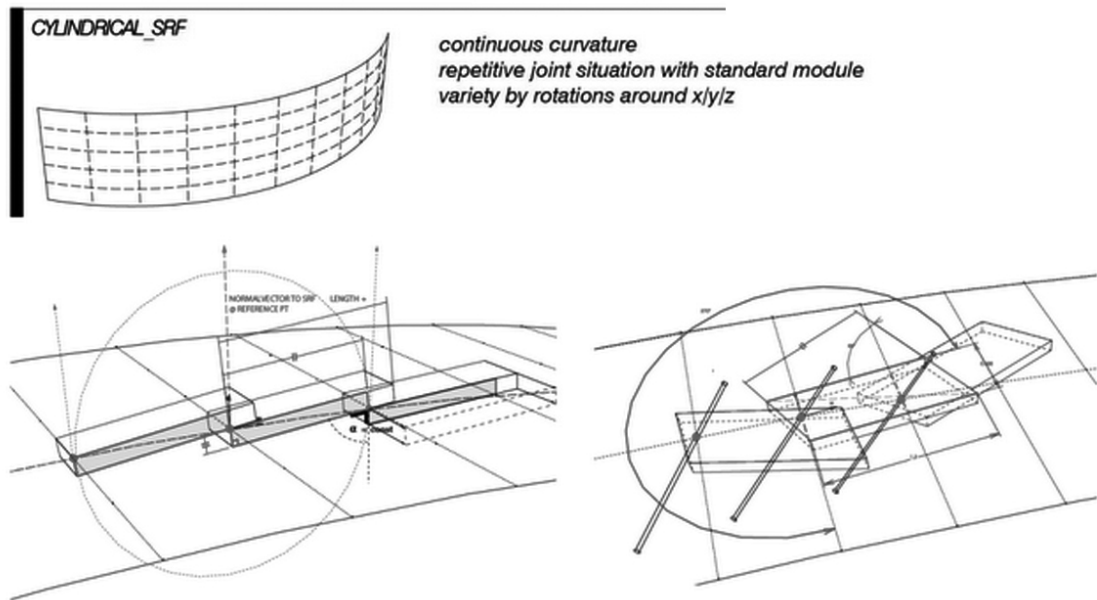


Figure 62 a : Typologie de calepinage en fonction de la nature des surfaces, ici pour une portion de cylindre

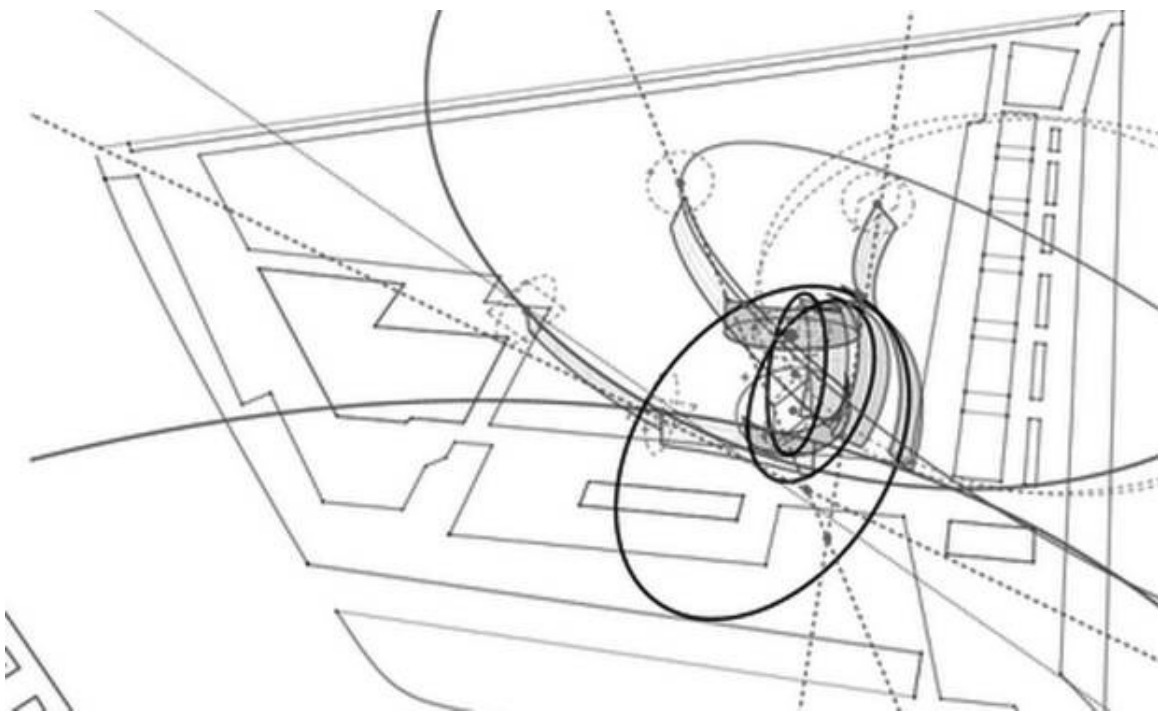


Figure 62 b : Dessin des surfaces du pavillon « according to urban parameters »



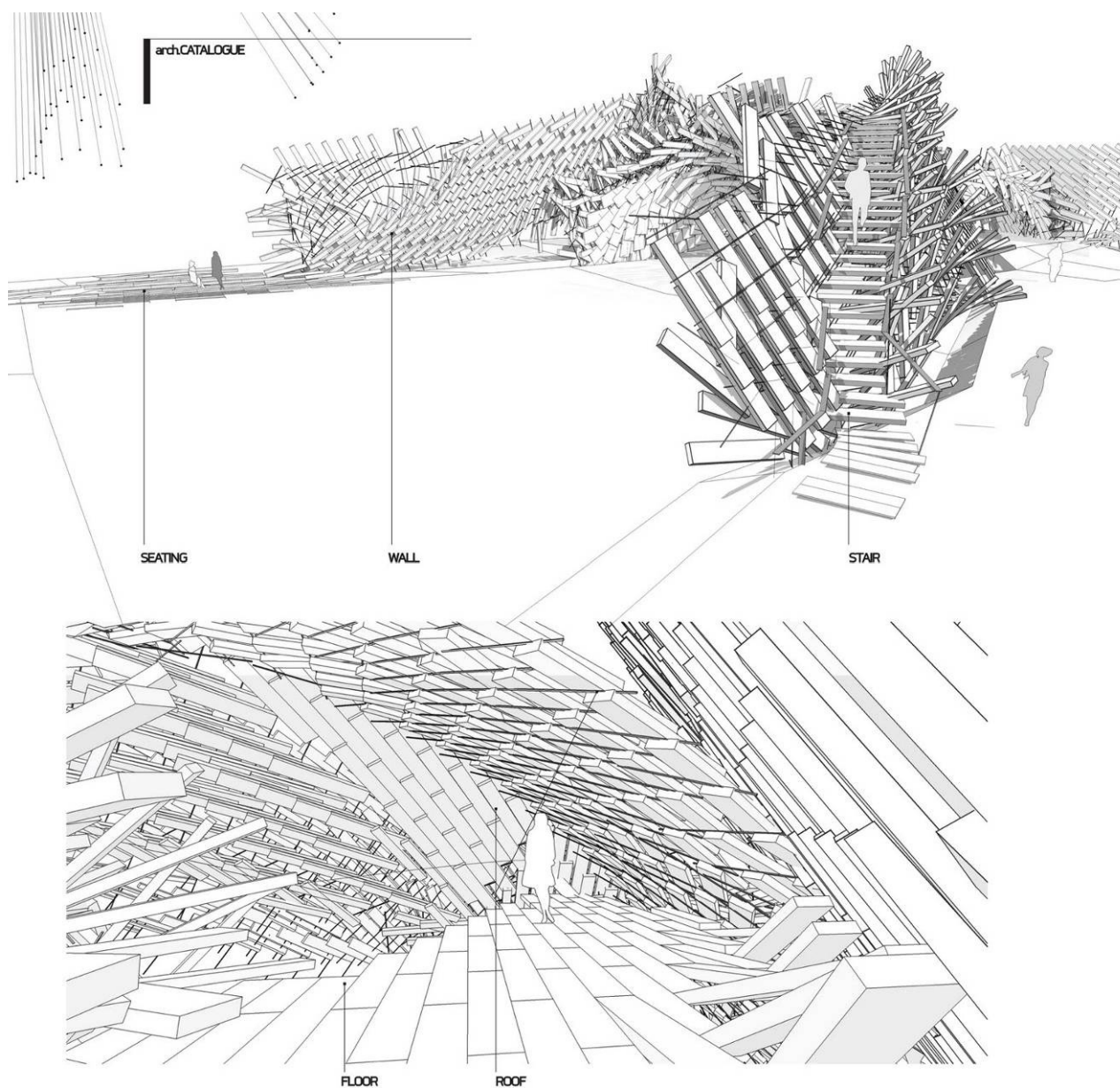


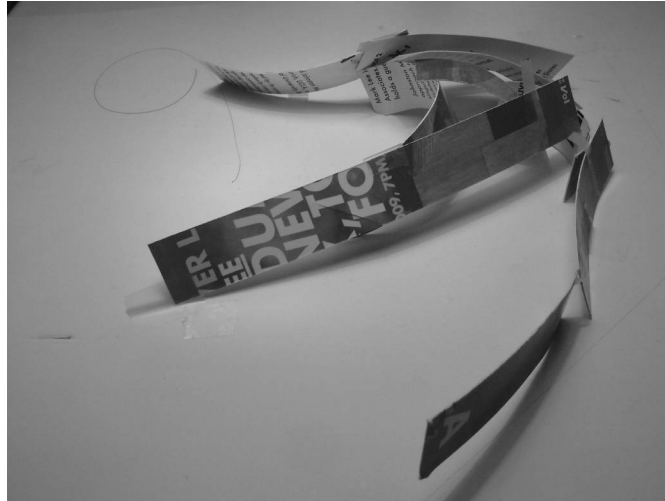
Figure 62c : Instanciation des calepinages sur les surfaces du pavillon

**Figure 62 : « Radical Reproduction » Daniela Kroenhert et Adam Orlinsky, CrossOver Studio (Universität Die Angewandte, Vienne)**

Le processus s'initie par la recherche de différents types de calepinage de tasseaux de bois (cf. Figure 62.a). Chaque type de calepinage est adapté à la nature des surfaces auxquelles les assemblages sont destinés (surfaces planes, cylindriques, coniques ou à double courbure inversée). Des évaluations de « surfaces types » grâce au moteur de calcul par éléments finis, intégré dans Digital Project ont été effectuées pour permettre aux étudiants de définir les contraintes s'appliquant sur les assemblages (cela peut être rapproché d'échelles géométrique et technique).

Parallèlement à ce premier temps, les étudiants ont conçu un ensemble de surfaces répondant au site (cf. Figure 62.b). Ces surfaces ont été conçues grâce à des expérimentations sur des maquettes physiques (cf. Figure 63). La conception de ces surfaces a été définie en fonction du voisinage et des

volumes de la place (*échelle de voisinage*<sup>67</sup> ou bien *échelle parcellaire*) et des flux des passages des piétons (*échelle fonctionnelle*<sup>68</sup>). Le site, s'il intervient dans la conception des surfaces (cf. entretiens, discours du jury et représentation de rendu), n'est pas pris en compte dans le modèle paramétrique : il est représenté (cf. Figure 62.b) mais n'est pas lié paramétriquement aux surfaces (par exemple une transformation du tracé du site n'influe pas sur les surfaces). Les géométries créées au moyen de maquettes physiques ont ensuite été traduites en portions de cylindre sur Digital Project (cf. Figure 62.b).



**Figure 63 : Photographie d'une maquette d'étude, source: Adam Orlinsky**

Le processus de conception se conclut par l'instanciation du système de calepinage pensé *a priori* sur les surfaces définissant le pavillon. Ce travail d'instanciation se compose de deux temps : une instanciation automatisée des typologies de calepinages (cf. Figure 64), puis une modification de certains assemblages en fonction des vues voulues (*échelle de visibilité*<sup>69</sup>), des usages (*échelle fonctionnelle*), de l'effet de fourmillement (« swarming ») souhaité (*échelle optique*<sup>70</sup>) et de l'ensoleillement (*échelle géométrique*<sup>71</sup> ou *géographique*<sup>72</sup>) (cf. Figure 62 c). L'instanciation n'est pas ici systématique, mais est le lieu d'opérations de conception variées.

<sup>67</sup> L'échelle de voisinage consiste à : « attribuer des mesures par contiguïté, relativement à des éléments appartenant au voisinage et induisant une continuité spatiale » (Boudon et al. 2000, p.171) .

<sup>68</sup> L'échelle fonctionnelle relève de l'activité de donner des mesures à un objet « en correspondance à quelque élément extérieur qui en règle la destination, l'utilisation, ou l'adapte à un usage » (Boudon et al. 2000, p.168).

<sup>69</sup> L'échelle de visibilité rend compte d'opérations cognitives visant à « situer un objet ou une partie d'objet de manière à ce qu'il soit vu d'un lieu ou qu'il ait vue sur un lieu » (Boudon, 2000. p. 174).

<sup>70</sup> L'échelle optique correspond à l'activité de « tenir compte d'un point de vue –au sens propre- pour donner à une partie ou à un tout de l'espace architectural une mesure qui fasse intervenir les modalités suivant lesquelles il est vu » (Boudon et al. 2000, p.167).

<sup>71</sup> L'échelle géométrique concerne les opérations cognitives qui impliquent de « recourir à des considérations géométriques pour découper ou référer des parties ou des totalités de l'espace architectural ou encore pour transcrire d'autres modalités de mesure » (Boudon 2000, p. 181).

<sup>72</sup> L'échelle géographique désigne l'attribution de mesures « à partir de considérations sur l'orientation des points cardinaux, la situation et le modelé du terrain, les données climatiques, etc. » (Boudon 2000, p. 173).

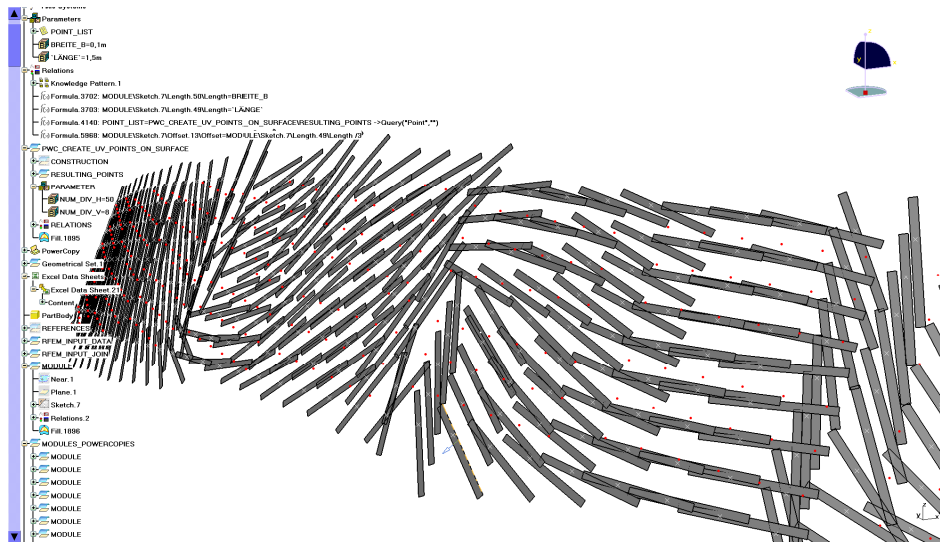
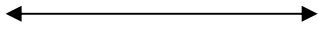
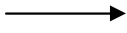
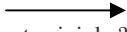


Figure 64 : capture d'écran du modèle paramétrique développé pour le projet "Radical Reproduction", source: Adam Orlinsky

Ces diverses opérations de conception identifiées sont formalisées dans le tableau suivant.

| Opérations de Conception Architecturale  | Relation  | Opérations de modélisation paramétrique  |
|--|---|--|
| La conception des calepinages relève principalement de mesures géométriques et techniques : elles sont liées aux capacités techniques des assemblages de tasseaux de bois et à la géométrie des surfaces.                                |  <p>-probable partage de <i>pertinences</i><br/>-structure d'opération dominée par l'<i>échelle technique</i></p> <p>On peut penser que savoir qu'une powercopy devait être créée a amené les étudiants à penser spécifiquement en terme de système et, d'entrées et de sorties claires.</p> | Les caractéristiques des calepinages sont modélisées dans des « powercopies » <i>via</i> la définition de règles de comportements de celles-ci.  |
| Pour la conception de ces surfaces le voisinage avec les volumes de la place (échelle de voisinage) et les flux des passages des piétons (échelle fonctionnelle) ont été pris en compte.   |  <p>Est suivi de</p> <p>Disjonction entre les opérations de modélisation et les opérations de conception du temps 2.</p>   | La modélisation des surfaces ne prend en compte que des pertinences techniques : elles sont représentées comme des portions de cylindres. Elles répondent donc à l'échelle technique mise en œuvre lors du temps 1, mais ne répondent pas aux échelles de voisinage et fonctionnelle mises en œuvre lors du temps 2. |
| Mise en place du calepinage en fonction des vues voulues (échelle optique), de rôles tels que « mur », « plafond », « banc » (cf. Figure 62c) (échelle fonctionnelle) en fonction des effets de « swarming » créés (échelle de modèle ?) |  <p>est suivi de ?</p> <p>Détermination de l'instanciation de la powercopy par les opérations de conception.</p>   | L'instanciation a eu lieu en fonction de la nature des surfaces (échelle technique), tel que défini lors du temps 1, mais aussi en fonction des opérations architecturales liées à la définition des effets voulus par le mouvement des tasseaux de bois (échelle optique, de modèle, fonctionnelle).                |

**Tableau 28 : Opérations de conception de l'architecture et du modèle paramétrique identifiées pour le projet "Radical Production"**

**- Peut-on observer des « unités » « niveaux » ou « moments » de conception ?**

Dans un premier temps, des opérations ont permis de donner de la mesure au système génératif de calepinage. Puis, dans un deuxième temps, des opérations cognitives de conception ont donné de la mesure à l'instance « pavillon » de ce système. L'analyse montre une différenciation explicite (et revendiquée) de la conception du système paramétrique par rapport à son instanciation. Il semble que nous soyons ici en présence de deux *unités de conception* : celle du modèle paramétrique et celle de l'instance. Il est intéressant de noter que les pertinences dont relèvent les deux *unités de conception* ne sont pas les mêmes. Dans le premier temps, la conception du système paramétrique relève principalement de mesures géométriques et techniques : elles sont liées aux capacités techniques des assemblages de tasseaux de bois et à la géométrie des surfaces. Dans le deuxième temps, l'instanciation de l'objet « pavillon » (temps 2 et 3) fait appel à des pertinences plus variées, liées à l'échelle humaine, aux usages, ainsi qu'à la vue sur le parc et à l'ensoleillement.

**- Peut-on observer des opérations de modélisation paramétrique spécifiques ?**

|                               |  |
|-------------------------------|--|
| Opérations de découpage       | <p>Dans les multiples niveaux de découpage possible on peut ici mettre en avant le découpage explicite des unités de conception et de modélisation</p> <p>La décision de discrétiser le modèle des calepinages par tasseaux de bois unique (et non par rangée par exemple) chacun possédant ses caractéristiques (orientation, point d'ancrage) peut être considéré comme une opération de découpage.</p> <p>Le modèle paramétrique n'étant pas disponible il n'est pas possible d'analyser très avant le découpage de celui-ci sur la seule base des captures d'écran</p> |
| Opérations de dimensionnement | <p>L'association des power-copies aux surfaces <i>via</i> des trames orthonormée de plans tangents est une mise en relation de la dimension calepinage avec les intentions architecturales (de vue, de fonctionnement du pavillon, <i>etc.</i>).</p>   |

**Tableau 29 : Opérations cognitives spécifiques à la conception du modèle paramétrique**

**-Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?**

Données non disponibles.

**- Peut-on observer des opérations de collaboration liées à la modélisation paramétrique ?**

Données non disponibles.



## 6 Analyse et caractérisation des opérations cognitives en jeu lors de l'usage de la modélisation paramétrique pour la conception architecturale

Ce Chapitre 6 vise à répondre à la première des deux questions soulevées par la recherche : « Quelles sont les opérations cognitives impliquées lorsque la conception de modèles paramétriques accompagne la conception architecturale ? »

La méthode d'interrogation des opérations cognitives en jeu lors de l'usage de la modélisation paramétrique en conception architecturale consiste à retrouver et interpréter des signes indiciaires d'*opérations de conception architecturale* et d'*opérations de conception de modèles paramétriques* (cf. Chapitre 5). Notre hypothèse de recherche est que ces opérations peuvent être énoncées en termes d'*opérations de la conception*, telles que définies en architecturologie (Boudon et al. 2000; Lecourtois 2006a; Ben Rajeb 2012).

La caractérisation des opérations cognitives de la conception architecturale fait l'objet d'une riche littérature architecturologique (Boudon et al. 2000; Lecourtois 2010). Nous nous intéressons ici plus particulièrement aux *opérations de conception de modèles paramétriques* en conception architecturale, à leurs mises en œuvre en conception architecturale et à leurs relations.

Dans un premier temps, nous questionnons les *opérations élémentaires* de la conception en architecturologie (*découpage*, *dimensionnement* et *référenciation*) en vue de saisir la manière dont elles peuvent rendre compte des mécanismes cognitifs de conception de modèles paramétriques (6.1). Par ailleurs, les concepts de « propriétés d'échelles » et de « relations d'échelles » nous permettent d'interroger les opérations de conception du modèle paramétrique (6.2) et, leurs liens aux opérations de conception architecturale (6.3).

Notre analyse a mis en évidence l'intervention d'opérations d'un autre ordre que des opérations *de conception*. Ces opérations influent sur le processus de conception et se rapprochent de ce que Ben Rajeb nomme des *opérations pragmatiques* (Ben Rajeb 2012) (6.2).

**Plan du chapitre :**

|            |  |            |
|------------|--|------------|
| <b>6.1</b> | <b>Conception de modèles paramétriques : Opérations cognitives élémentaires.....</b>   | <b>195</b> |
| 6.1.1      | Opération de découpage   | 195        |
| 6.1.2      | Opération de dimensionnement de modèles paramétriques  | 197        |
| 6.1.3      | Opération de référencement   | 202        |
| <b>6.2</b> | <b>Spécificité de la mise en œuvre des opérations conception de modèles paramétriques</b>                                    | <b>204</b> |
| 6.2.1      | Concevoir le processus de génération ou concevoir l'instance : Unités de conception, niveaux de conception ?                 | 204        |
| 6.2.2      | Propriétés d'échelles spécifiques à l'usage de la modélisation paramétrique  | 208        |
| <b>6.3</b> | <b>Relations entre opérations de conception du modèle paramétrique et opérations de conception architecturale .....</b>      | <b>210</b> |
| 6.3.1      | Relations temporelles : opérations en <i>relais</i> ou <i>cascade</i>  | 211        |
| 6.3.2      | Relation dans l'espace : juxtaposition, partage de référence et/ou de pertinence   | 214        |
| <b>6.4</b> | <b>Opérations tierces impliquées .....</b>   | <b>218</b> |
| 6.4.1      | Opération pragmatique de collaboration   | 219        |
| 6.4.2      | Opérations pragmatiques d'usage de l'outil   | 222        |
| 6.4.3      | Opérations logiques  | 226        |
| <b>6.5</b> | <b>Synthèse des caractérisations cognitives de la conception de modèles paramétriques en conception architecturale .....</b> | <b>231</b> |



## 6.1 Conception de modèles paramétriques : Opérations cognitives élémentaires

Dans ce paragraphe, nous interrogeons la possibilité de décrire des *opérations élémentaires de la conception de modèles paramétriques*, à partir de concepts définis par l'architecturologie (Boudon et al. 2000, p.154). En architecturologie fondamentale (cf. 5.1), la conception est une activité complexe (Lecourtois 2012) que trois *opérations élémentaires* permettent d'éclairer. Ces opérations sont : l'opération de *découpage* (0), l'opération de *dimensionnement* (6.1.2) et l'opération de *référenciation* (6.1.3). Dans la réalité, ces *opérations élémentaires* peuvent être isolées ou conjointes.

### 6.1.1 Opération de découpage

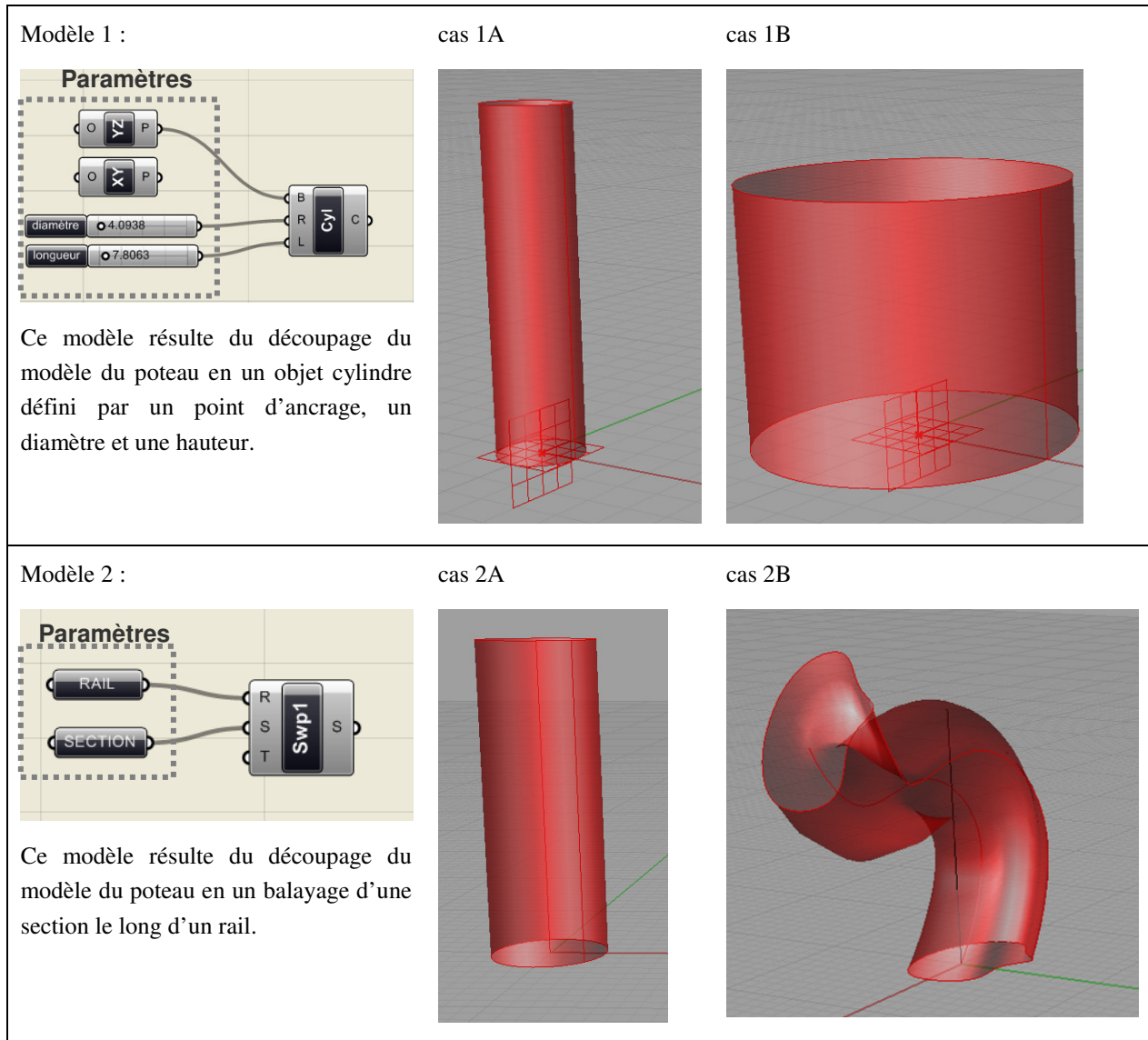
L'opération de *découpage* est l'opération par laquelle le concepteur décide de ce sur quoi va porter sa mesure : c'est-à-dire de la (ou des) *dimension(s)* (Boudon 2001, p. 114). Si l'on reprend l'exemple de la conception d'une piscine, quand le concepteur décide de s'intéresser à la largeur d'un bassin, il opère un « découpage » mental du projet par lequel il isole la *dimension* « largeur du bassin » en vue de lui attribuer des mesures.

On peut observer que l'opération de *découpage* est également mise en œuvre lors de la conception de modèles paramétriques. Un concepteur « découpe » alors mentalement le modèle paramétrique qu'il est en train de concevoir pour identifier ses *dimensions*. Par exemple, lorsqu'un concepteur décide de concevoir un modèle paramétrique où une figure géométrique est répétée sur une trame de points, il découpe mentalement son modèle en distinguant une *dimension* « figure géométrique » d'un côté, d'une *dimension* « trame de point » de l'autre. Dans l'exemple de la section 4.1.2 d'une application du pattern *controller* (cf. 4.1.2), une trame de cercles était définie pour que les rayons des cercles dépendent de leur distance à un point. Ici le modèle a été découpé pour isoler les rayons des cercles, en vue, d'ensuite lier leur définition avec le point « contrôleur ».

Pour une même géométrie produite, plusieurs *découpages* sont possibles. Une multiplicité de *découpages* peut aussi être mise en œuvre dans un modèle paramétrique pour obtenir les mêmes résultats (objets géométriques ou valeurs numériques). Les opérations de *découpage* sont liées aux points de vue portés sur le modèle par le concepteur ainsi qu'aux transformations voulues. Par exemple, pour modéliser une trame de poteaux, plusieurs *découpages* sont possibles. La trame de poteaux peut être modélisée comme la copie d'une primitive cylindre sur chacun des points de la trame (cf. Figure 65 cas 1A) ou comme un balayage de profil (cf. Figure 65 cas 2A). Ces deux cas sont le résultat de *découpages* différents : *découpage* du modèle paramétrique du poteau à partir d'une forme unique pour le premier, *découpage* du modèle du poteau comme d'une courbe et d'un axe pour le second. Dans un premier temps, ces deux *découpages* donnent lieu à des géométries identiques. Mais les paramètres, c'est-à-dire les variables indépendantes du modèle paramétrique (cf. Chapitre 2), définissent des domaines de solutions différents (cf. Figure 65 B). Le *découpage* du modèle paramétrique induit des paramètres spécifiques (cf. Figure 65 modèles 1 et 2) qui définissent à leur tour des domaines de solutions.

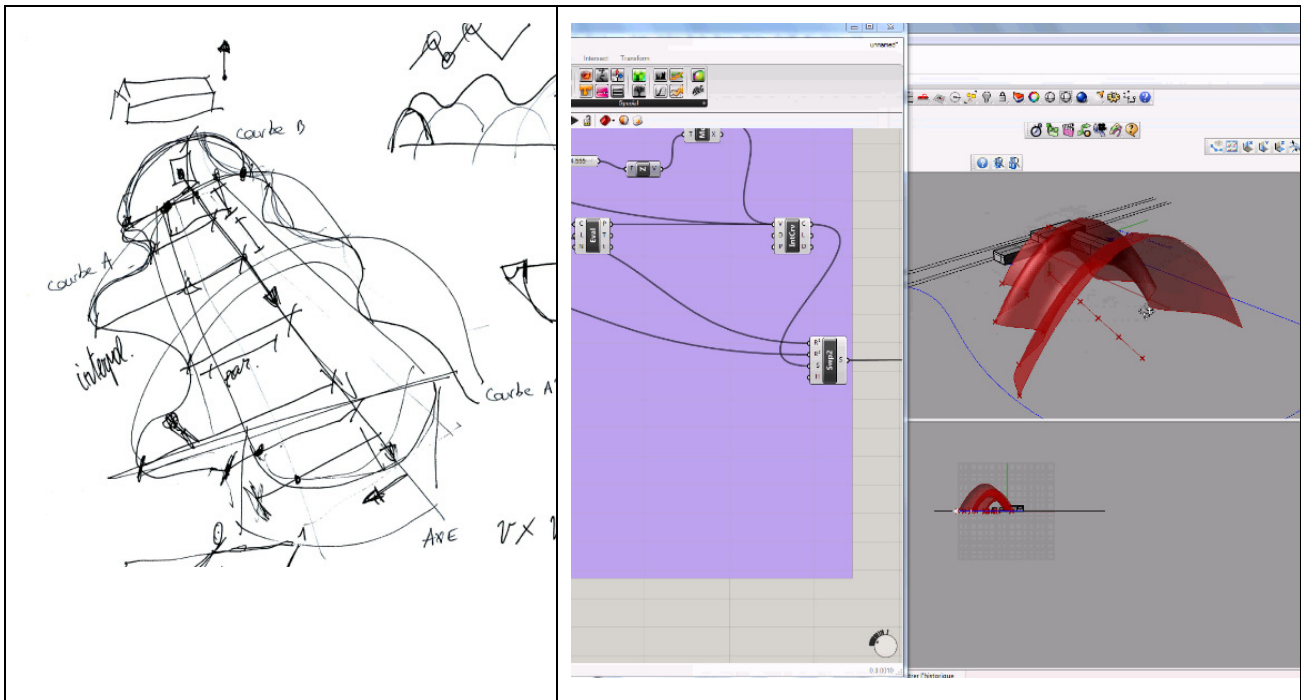
Cette *opération élémentaire de découpage de modèle paramétrique* est, à ce point de l'analyse, susceptible d'être liée à des opérations de conception architecturale. Dans l'exemple de la trame de

poteaux, si le concepteur estime que le modèle de poteau est susceptible d'évoluer en fonction du diamètre ou de la hauteur, il peut le modéliser sous la forme d'une primitive cylindre (cf. Figure 65 cas 1B). S'il estime qu'au cours de l'évolution du projet, le modèle paramétrique risque plutôt d'évoluer du point de vue de la section du poteau en fonction du choix du coffrage, du revêtement prévu, ou d'autres considérations, alors mieux vaut construire le modèle à partir d'un balayage de profils (cf. Figure 65 cas 2B).



**Figure 65: Comparaison de deux modèles paramétriques (développés sur Grasshopper) impliquant des découpages différents mais des instances semblables**

On peut prendre un autre exemple d'opération de *découpage* du modèle paramétrique dans le projet de Piscine de Sebastiaan Leenknecht et Rebecca Paulin (groupe A2, Expe1, cf. annexe p.393). Dans ce cas, les étudiants ont conçu un modèle paramétrique dont la géométrie repose sur un paramètre principal : l'axe de leur bâtiment. Les modèles du bassin et de la couverture du projet sont situés dans l'espace de la scène Rhinocéros explicitement grâce au segment de droite formalisant cet axe (cf. Figure 66). Ce segment est une variable de référence : s'il est modifié, toute la géométrie du projet se réoriente automatiquement par propagation (cf. 2.3.1).



**Figure 66: Esquisses préparatoires à gauche et extrait du processus de conception de modèle à droite, Projet groupe A2, Expe1, sources Sebastiaan Leenknecht et Rebecca Paulin**

Comme dans le précédent cas, l'opération de *découpage* du modèle paramétrique observée paraît entremêlée avec la conception architecturale. Ce *découpage* de l' « axe du projet » est clairement valorisé dans le cadre de la conception architecturale pour définir l'implantation du projet, contrairement par exemple à l'emprise de la parcelle ou aux bâtiments existants qui ne sont pas directement pris en compte dans cette opération.

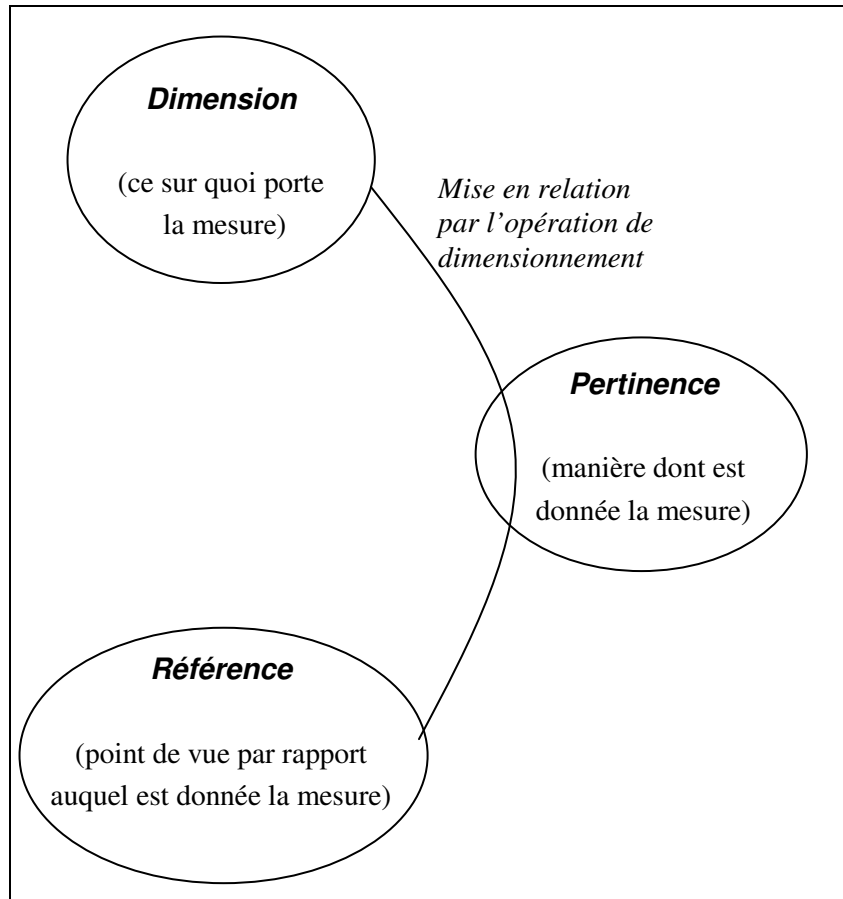
L'opération élémentaire de conception appelée *découpage* permet bien d'interroger certaines opérations de conception de modèles paramétriques.

### 6.1.2 Opération de dimensionnement de modèles paramétriques

Dans la modélisation paramétrique, une opération majeure identifiée par la plupart des acteurs est la mise en relation d'éléments préalablement distingués. « Mettre en relation » est souvent considérée comme la plus importante activité de la modélisation paramétrique. C'est par ailleurs la raison pour laquelle la modélisation paramétrique est aussi appelée « géométrie associative ». Hugh Whitehead écrit ainsi (Whitehead in Woodbury 2010): « *the first requirement [for parametric modelling] is an attitude of mind that seeks to express and explore relationships* ». Cette opération peut être par exemple : associer des variables, relier une fonction avec des données extérieures, associer une géométrie à une contrainte, etc.

Cette « mise en relation » semble proche de l'opération élémentaire de dimensionnement proposée par l'architecturologie. En effet, en architecturologie, l'opération de dimensionnement n'est autre que la mise en relation d'une *dimension* (ce sur quoi porte la mesure) avec un *espace de référence* (ce par rapport à quoi est donnée la mesure) via une *pertinence* (façon dont est donnée la mesure) (Boudon et al. 2000, p.119-123). Cette mise en relation est schématisée dans la Figure 67. Si l'on reprend l'exemple

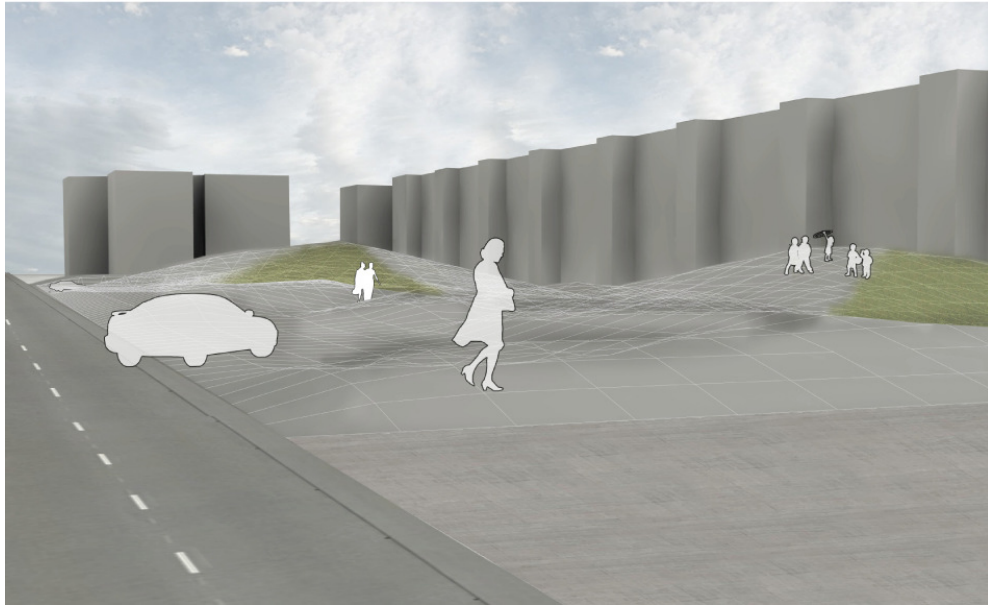
de la conception de piscine du paragraphe précédent, lorsqu'un concepteur décide d'attribuer une mesure à la largeur d'un bassin (intervention de la *dimension*) par rapport à un certain confort des nageurs (intervention de l'*espace de référence*) par exemple pour que ceux-ci puissent nager à douze sans se gêner (intervention de la *pertinence*), la mise en relation de la largeur du bassin avec le confort des nageurs est une opération mentale de *dimensionnement*.



**Figure 67: Proposition de schématisation de l'opération de *dimensionnement* comme mise en relation, d'après (Boudon et al. 2000)**

On retrouve également des opérations de *dimensionnement* dans la conception de modèles paramétriques.

Prenons un exemple issu du projet « Topographies » des étudiants Aurea Rodriguez, Pablo Gancedo, Samya Pelloquin et Mathias Saboya (cf. enseignement « Architecture paramétrique » de l'ENSA de Marne la Vallée, analyse complète insérée en annexe p.374). Lors de ce processus de projet, les concepteurs souhaitaient travailler sur l'espace d'un parking de la cité des 3000, situé en région parisienne, pour créer une nouvelle « topographie » de cet espace public, pouvant animer l'espace et remplir plusieurs fonctions (espaces de jeu, de circulation, de parking, etc.) (cf. Figure 68).



**Figure 68 : Extrait de la présentation du projet « Topographies » d’Aurea Rodriguez, Pablo Gancedo, Samya Pelloquin et Mathias Saboya**

Le modèle paramétrique conçu lors de ce projet reprend le *découpage* relatif à l’emprise du parking et s’appuie sur une surface de référence modélisée dans Rhinocéros. Les altitudes de cette surface sont transformées en vue de générer la « topographie » visée. Nous nous intéressons ici au *dimensionnement* de ce modèle paramétrique de surface.

Le *dimensionnement* du modèle paramétrique de la surface est opéré par la définition d’une trame de points, générée sur la surface de référence (cf. Figure 69A), dont les coordonnées en Z sont manipulées. Ces points, aux altitudes générées par le modèle paramétrique, contrôlent la surface de « Topographie » visée (cf. Figure 69B).



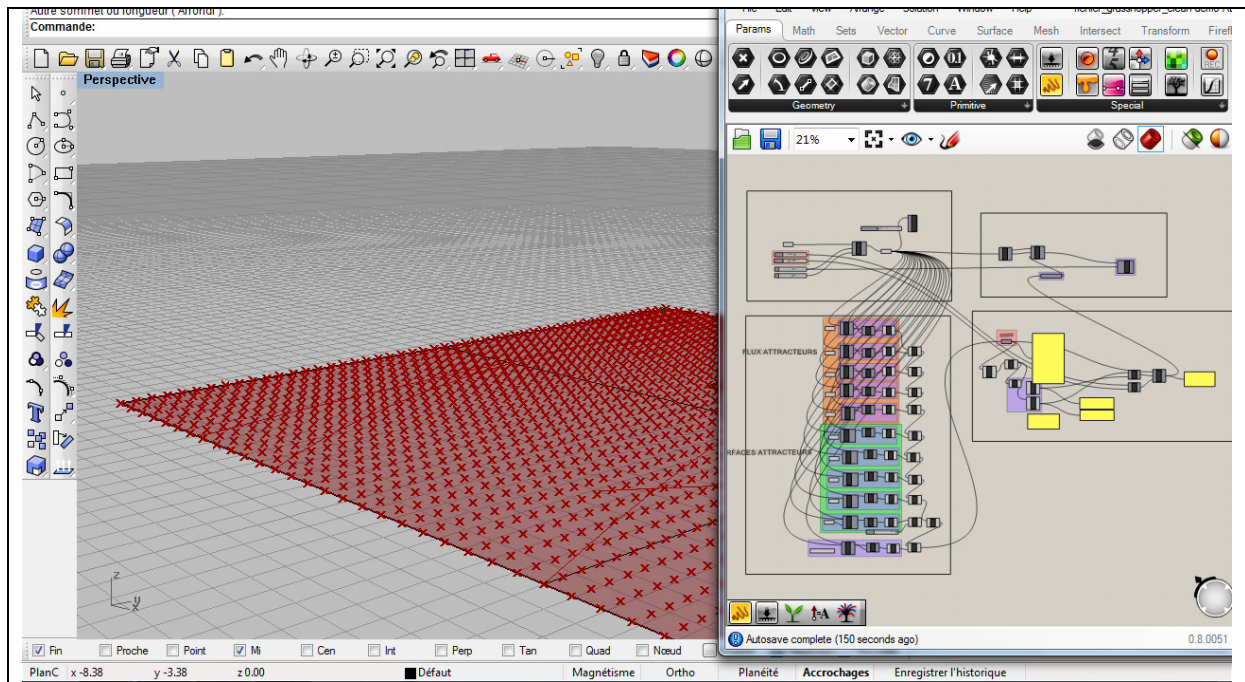


Figure 69 A : Surface de référence et trame de points à une altitude 0

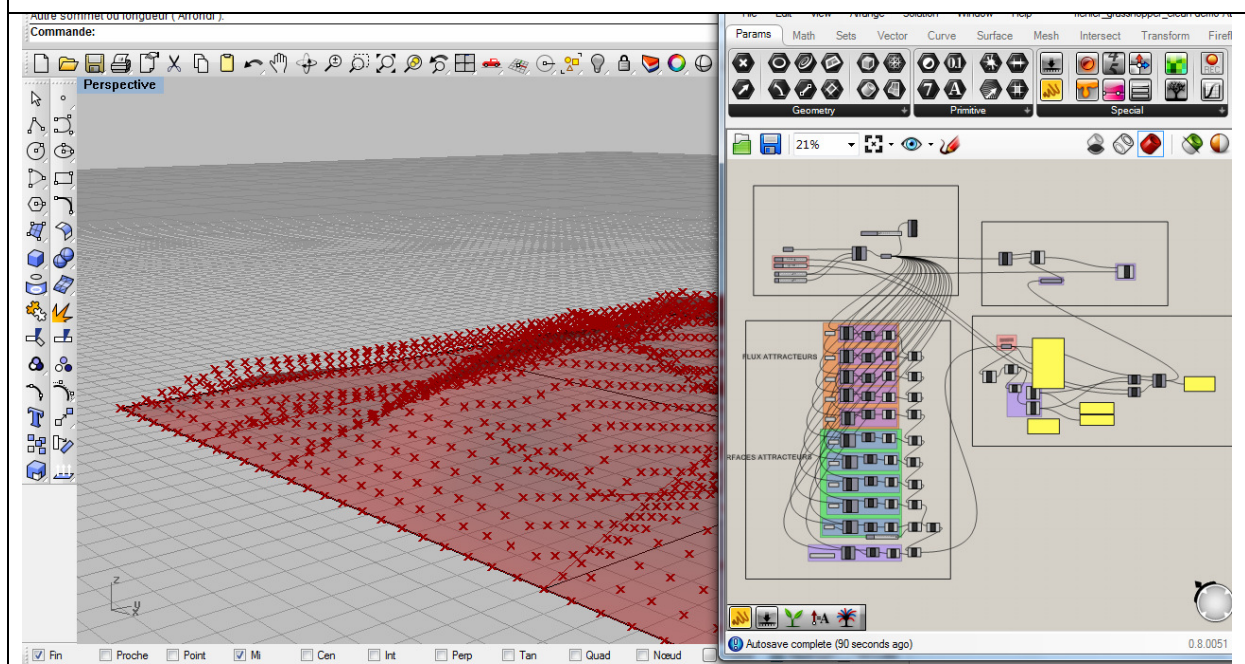


Figure 69 B : Surface du projet avec modulation des altitudes des points

**Figure 69:** Captures d'écran du modèle paramétrique du projet « Topologies » développé par Aurea Rodriguez, Pablo Gancedo, Samya Pelloquin et Mathias Saboya, source de l'image : Aurélie de Boissieu.

Les altitudes des points de la surface sont évaluées en fonction de la position de zones de circulation dont les altitudes doivent rester à zéro. Ces zones de circulation sont formalisées par des courbes. Dans le modèle paramétrique, plus un point de la surface est proche d'une courbe, plus son altitude est faible. En termes architecturologiques, ces points de contrôle de la surface sont *dimensionnés* en fonction de leur distance à des courbes. Ce *dimensionnement* du modèle paramétrique s'entremêle avec une

opération de conception architecturale, visant à rendre la « Topographie » praticable par des piétons. Cette opération de conception architecturale pourrait relever d'une *échelle fonctionnelle* : elle donne de la mesure à un objet « *en correspondance à quelque élément extérieur qui en règle la destination, l'utilisation, ou l'adapte à un usage* » (Boudon et al. 2000, p.168).

Dans le modèle paramétrique, les altitudes des points de la surface (cf. Figure 69 B) sont également *dimensionnées* par rapport à un maximum et un minimum définis au moyen de paramètres. C'est-à-dire que toutes les valeurs des altitudes des points de contrôle sont traitées<sup>73</sup> dans le modèle pour être comprises dans des extremums définis par le concepteur. Cette nouvelle opération de *dimensionnement* du modèle paramétrique s'entremêle une fois de plus avec la conception architecturale. En effet ces extremums sont liés à l'insertion globale de la « topographie » dans le site. L'altitude maximum de la topographie est contrainte par le voisinage des habitations : la « topographie » ne doit pas obstruer les vues des appartements situés au premier étage de la cité. Cette opération de conception architecturale pourrait relever de la mise en œuvre d'une *échelle optique*<sup>74</sup>, d'une *échelle de voisinage*<sup>75</sup>.

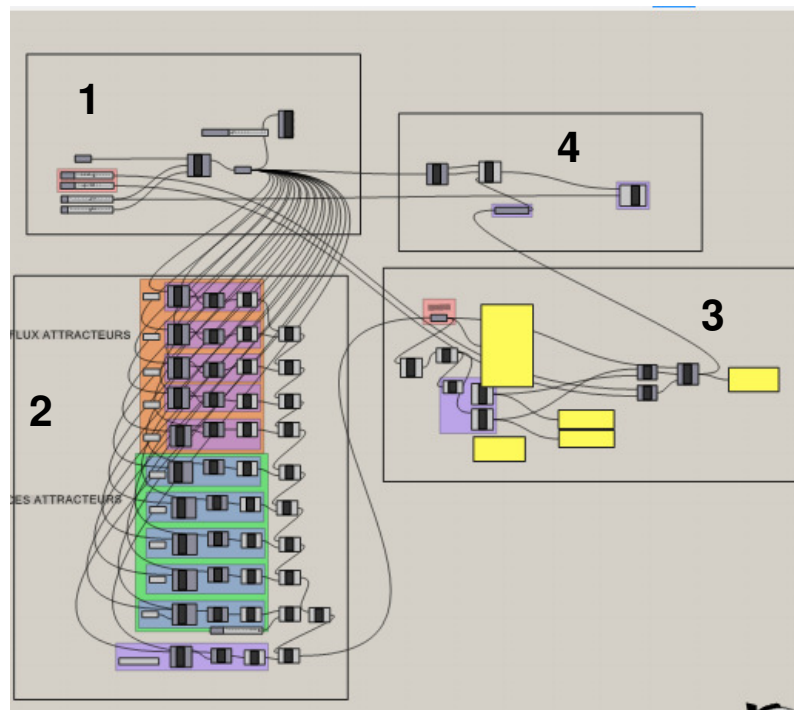


Figure 70 : Image du modèle paramétrique du projet « Topographie »

Finalement le modèle paramétrique de ce projet s'organise en quatre étapes (cf. Figure 70) :

- 1- une trame orthogonale de points est créée sur la surface de référence ;
- 2- la distance de chacun des points de contrôle de la surface à la courbe représentant le chemin le plus proche est calculée ;

<sup>73</sup> Le traitement utilisé relève d'un des Patterns de Woodbury : le pattern de « Mapping ».

<sup>74</sup> C'est à dire d'une opération mentale visant à « *tenir compte d'un point de vue –au sens propre- pour donner à une partie ou au tout de l'espace architecturale une mesure faisant intervenir les modalités suivant lesquelles elle va être vu* » (Boudon et al. 2000, p.175)

<sup>75</sup> Opération mentale visant à « *attribuer des mesures par contiguïté, relativement à des éléments appartenant au voisinage et induisant une continuité spatiale* » (Boudon et al. 2000, p.171)

3- ces valeurs (une valeur de distance par point) sont transformées pour conserver leur répartition mais changer d'extrémums, ici la valeur maximale est définie par le concepteur et la valeur minimale est zéro pour correspondre au niveau du sol ;

4- les valeurs finales obtenues sont utilisées pour générer les altitudes des points de contrôle de la surface finale.

Dans cet exemple, on a pu observer différentes opérations de *dimensionnement* du modèle paramétrique. Le formalisme de l'architecturologie nous permet donc ici de décrire des aspects de la conception de modèles paramétriques et, à partir de ces éléments de soulever d'autres questions : ici la question de l'entremêlement entre opération de conception du modèle paramétrique et opération de conception architecturale. Cet entremêlement, nous l'avons déjà soulevé lors de l'analyse de cas de description d'opérations de *découpage* de modèle paramétrique.

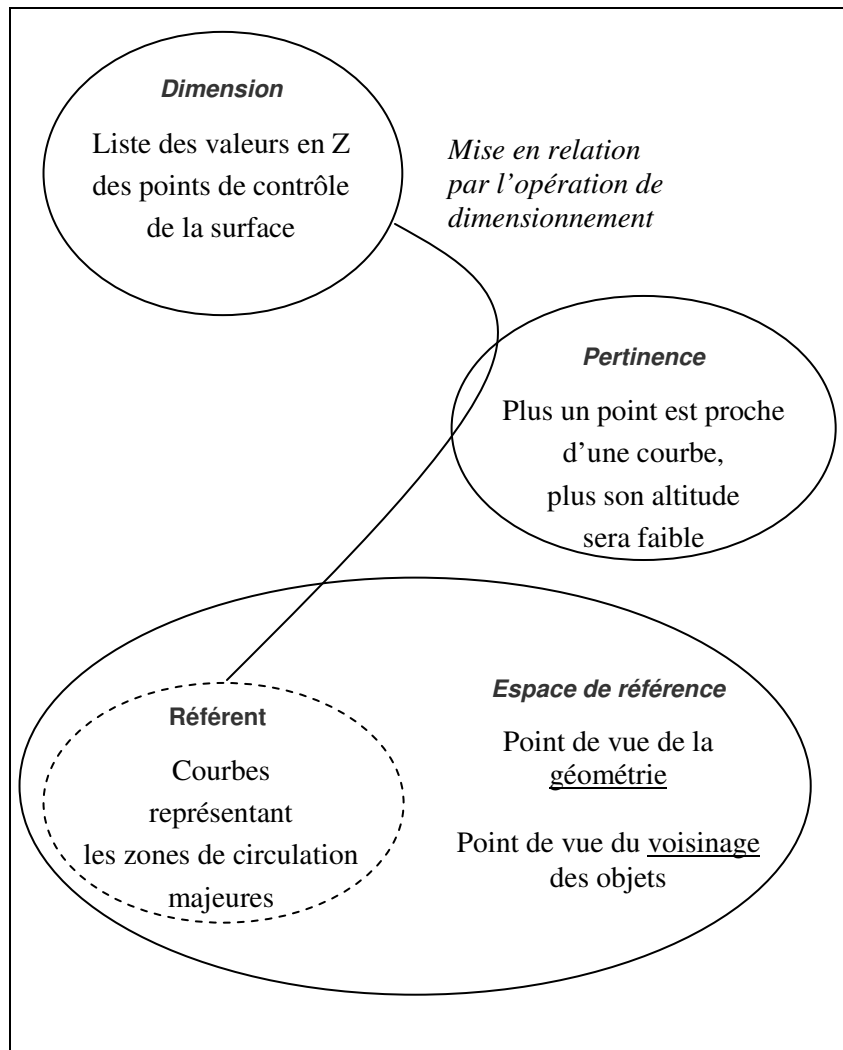
### 6.1.3 Opération de référenciation

En architecturologie, un *espace de référence* est une classe de représentations de la réalité, c'est-à-dire un point de vue personnel sur la réalité ou sur un objet référent. Par exemple, à partir d'un même objet référent « parcelle », un concepteur peut retenir différents *espaces de références* : la morphologie de la parcelle, sa géographie, son inscription socioculturelle, etc. (Boudon et al. 2000, p.111). L'*espace de référence* est opératoire dans la conception.

L'*opération de référenciation* est l'action par laquelle le concepteur se réfère à un *espace de référence*, un objet réel, ou par laquelle un *espace de référence* est associé à une *dimension*. Il semble que cette *opération élémentaire* de la conception permette d'interroger la conception de modèles paramétriques.

Si l'on reprend l'exemple du projet « Topographie » vu dans la section précédente, la conception du modèle paramétrique des points de contrôle de la surface semble faire appel au moins à un *espace de référence* géométrique et, pourquoi pas, à un *espace de référence* renvoyant au *voisinage* des objets entre eux (cf. Figure 71).





**Figure 71 : Deux espaces de référence mises en œuvre dans la conception du modèle paramétrique du projet "Topographies"**

Selon Philippe Boudon, par la *référenciation* le concepteur « *installe tout ou une partie du projet dans un univers ayant un sens* », ce qui permet ensuite au concepteur de repérer des possibilités de pertinence (Boudon et al. 2000, p.111). L'espace de référence indique donc dans quel univers de possibilités se situe le concepteur au moment où il donne de la mesure. Dans le cas des opérations de conception de modèle paramétrique, si les *espaces de référence géométrique* et *technique* sont les plus courants, ce ne sont pas les seules références sollicitées. L'entremêlement précédemment observé entre opérations de conception du modèle paramétrique et opérations de conception architecturales se retrouve dans le cas des *opérations de référenciation* de l'une et de l'autre, dans lesquelles on observe des porosités et des échanges de l'une vers l'autre. Ces échanges ou porosités sont analysés plus précisément dans la suite de la thèse : elles ne semblent pas relever de la conception elle-même mais d'opérations tierces, opérant le passage d'une conception à l'autre.

## **6.2 Spécificité de la mise en œuvre des opérations conception de modèles paramétriques**

La section précédente interroge la possibilité de décrire des *opérations élémentaires* de la conception de modèles paramétriques en les isolants. La section suivante revient sur les agencements possibles de ces opérations.

### **6.2.1 Concevoir le processus de génération ou concevoir l'instance : Unités de conception, niveaux de conception ?**

Au sein des travaux théoriques qui interrogent l'usage du numérique en architecture, certains auteurs font le constat radical d'un changement de paradigme de la conception architecturale (Terzidis 2006; Kolarevic 2005; Migayrou 2003; Cache 1997). Ils avancent que l'objet de la conception n'est plus l'objet architectural mais le processus de conception lui-même (Deleuze 1988; Migayrou 2003).

Nous l'avons vu en effet dans les spécificités des mécanismes de la modélisation paramétrique (cf. Chapitre 2). Lorsqu'il conçoit un modèle sur un modèleur paramétrique, un concepteur manipule une représentation *symbolique*, chaîne de dépendance d'objets, définissant un processus génératif d'un domaine de solution, et une représentation *explicite*, instance du système, permettant d'en visualiser la géométrie. Quels mécanismes cognitifs impliquent cette spécificité technique des modèleurs?

Nous avons interrogé jusqu'ici les concepts architecturologiques permettant d'interroger la conception en termes d'*opérations élémentaires*. Nous interrogeons à présent les concepts permettant de décrire les propriétés des *opérations élémentaires* (récurrentes, principales ? etc.) et de leur relation (plusieurs opérations portent-elles sur le même objet ? etc.).

En particulier, nous interrogeons les concepts d'*unité de conception* et de *niveau de conception* pour tenter d'explicitier les mécanismes en jeu lors de la conception de modèles paramétriques en particulier, par rapport à la distinction entre : concevoir un modèle paramétrique générique, concevoir un modèle paramétrique spécifique à un projet architectural et, concevoir une instance géométrique d'un modèle paramétrique.

#### **-Unité de conception**

Le concept d'*unités de conception* rend compte de l'activité d'un concepteur lorsque celui-ci choisit de donner des mesures à un ensemble de *dimensions* en même temps : par exemple quand deux façades sont dessinées ensemble, ou quand la conception va s'intéresser spécifiquement à une zone de la parcelle, etc. (Boudon et al. 2000, p.217).

Lors de la conception de modèles paramétriques, l'*unité de conception* pourrait rendre compte de l'activité d'un concepteur lorsque celui-ci découpe son modèle par exemple pour donner de la mesure à un sous-ensemble du modèle. Si l'on reprend de nouveau l'exemple de la conception du modèle paramétrique du projet « Topographie », quand les concepteurs se sont attachés à concevoir le code permettant de changer les extrêmes du groupe de valeurs « distances » en vue de les utiliser comme altitudes de la surface, ils ont mis en œuvre alors une *unité de conception*. Dans ce modèle paramétrique, il semble qu'au moins quatre unités de conception puissent être identifiées (cf. Figure 70).

### ***-Niveau de conception***

Nous l'avons vu, en architecturologie chaque opération de conception se situe par rapport à un point de vue appelé *espace de référence* (cf. 6.1.3). Lorsqu'un concepteur s'intéresse à un point de vue, il peut alors rapprocher des éléments qui sinon étaient sans rapport. Par exemple, si un concepteur s'intéresse à son projet du point de vue des usages de l'espace, peut-être va-t-il rapprocher dans sa réflexion l'espace d'accueil, l'emprise urbaine de la parcelle, la distribution des espaces, etc. Cette association mentale d'opérations de conception liées par un même point de vue est appelée « niveau de conception » (Boudon et al. 2000, p.210).

Les analyses de cas effectuées dans cette présente thèse montrent que la conception de modèles paramétriques implique trois récurrences d'activités, impliquant des *opérations de conception* distinctes. Ces activités sont (cf. Figure 72) :

- **l'activité de conception de Méta-Modèles paramétriques**: celle-ci implique la conception d'environnements de la modélisation paramétrique. C'est en quelque sorte la conception (dans le sens de mise en place en pensée) d'un « espace de la modélisation paramétrique ». Par exemple, c'est l'activité de David Rutten quand il conçoit et développe le modèle paramétrique Grasshopper. C'est également l'activité de conception de nouvelles fonctionnalités pour des modèles déjà développés (comme le plugin GHowl de Grasshopper) et, la conception de ressources (cf. 4.2) ou de méthodes (cf. 4.1) pour la modélisation paramétrique.

Cette activité peut être interrogée en termes d'*unité de conception*. En effet, dans ces activités se sont des sous-ensembles de modèles paramétriques qui sont conçus, voire des environnements ou des fonctionnalités à part entières.

- **l'activité de conception de modèles paramétriques** : Cette activité de conception implique le recours à un méta-modèle numérique (aux possibilités ouvertes par un modèle par exemple) pour la conception d'un modèle paramétrique. La plupart des cas issus du corpus rassemblé dans cette thèse relève de cette activité de conception.

De nouveau, cette activité semble pouvoir relever d'une *unité de conception* spécifique : c'est bien de la conception d'un modèle paramétrique spécifique qu'il s'agit, celui-ci pouvant faire intervenir plusieurs points de vue différents, c'est-à-dire plusieurs *niveaux de conception*.

- **l'activité de conception d'instanciation** : Cette activité vise à penser et construire un ou plusieurs objets particuliers à partir du domaine de solutions défini par un modèle paramétrique. Instanciations et modèles paramétriques peuvent être, ou non, conçus par les mêmes acteurs, dans le cadre, ou non, de mêmes processus de conception. Par exemple, dans le cas de l'usage de la modélisation paramétrique par Foster and Partners pour le projet du City Hall de Londres (cf. analyse détaillée en annexe p.386), l'équipe de conception architecturale a, sur certaines périodes, manipulé un modèle paramétrique développé par des acteurs du Specialist Modelling group en vue d'en explorer les instances possibles pour concevoir la forme finale du projet.

De nouveau, cette activité semble pouvoir prendre la forme d'une *unité de conception*, dont les dimensions sont les variables indépendantes (c'est-à-dire les paramètres) d'un modèle paramétrique déjà conçu.

Ces trois activités sont ici artificiellement distinguées pour en faire une première description. Les porosités et les renvois d'une activité à l'autre seront interrogés en termes d'opérations tierces dans la section 6.4.

| <i>Conception concernée</i>                            | <i>Activité de conception</i>   | <i>Concepteurs</i>   |
|--|---------------------------------|--|
| Conception de l'espace de la modélisation paramétrique | <b>Méta-Modèle paramétrique</b> | les développeurs de logiciels, le chercheur                                  |
| Conception d'un espace de modélisation paramétrique    | <b>Modèle paramétrique</b>      | les équipes spécialisées : le Specialist Modelling Group, Gehry Technologies |
| Choix de valeurs, instanciation                        | <b>Instance</b>                 | l'architecte   |

**Figure 72 : Les différentes unités de conception de la modélisation paramétrique**

On observe la mise en œuvre de ces activités impliquant des *unités de conception* particulières dans le projet « Radical Production » de Daniela Kroenhert et Adam Orlinsky du CrossOver Studio (cf. Annexe p. 353). Dans ce projet, les étudiants organisent de leur processus de conception en plusieurs temps, portant chacun sur des *unités de conception* spécifiques. Le processus de projet commence par l'exploration de différents calepinages de tasseaux de bois (cf. Figure 73.a) conçus en fonction de la nature des surfaces auxquelles les assemblages sont destinés (surfaces planes, cylindriques, coniques ou à double courbure inversée). Des modèles paramétriques de ces calepinages sont conçus en utilisant des powercopies. Les « powercopies » sont spécifiques au logiciel CATIA (cf. 2.3.3). Ce sont des « copies optimisées » que le concepteur définit en explicitant les entrées et les sorties de ces fragments de modèles. Ces « copies optimisées » peuvent ensuite être insérées dans différents contextes et s'adapter en fonction de ceux-ci. En parallèle, les étudiants conçoivent un ensemble de surfaces répondant au site (cf. Figure 73.b). En particulier, ces surfaces suivent les principaux flux de circulation piétonne sur la place et ménagent des vues privilégiées sur l'église faisant face au pavillon. Ces surfaces sont modélisées sur Digital Project sous la forme de portions de cylindres. Le système de calepinage pensé *a priori* est mis en œuvre à partir des intentions de surfaces. Cette mise en œuvre se développe en deux temps : une instanciation automatisée des typologies de calepinages sur les surfaces conçues, puis une modification de certains assemblages en fonction de vues, d'usages et d'ensoleillement recherchés (cf. Figure 73.c).

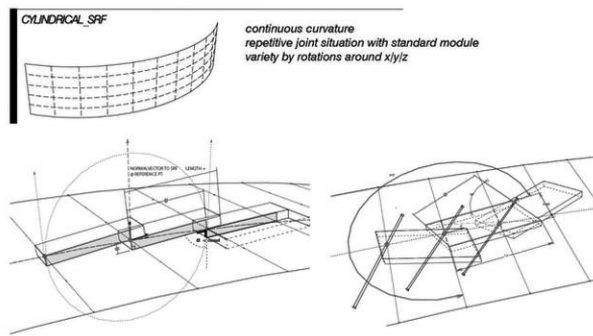


Figure 73.a : Typologie de calepinage en fonction de la nature des surfaces, ici pour une portion de cylindre

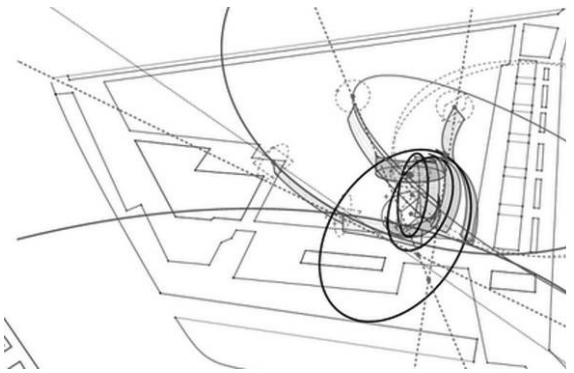


Figure 73.b : Dessin des surfaces du pavillon « according to urban parameters »

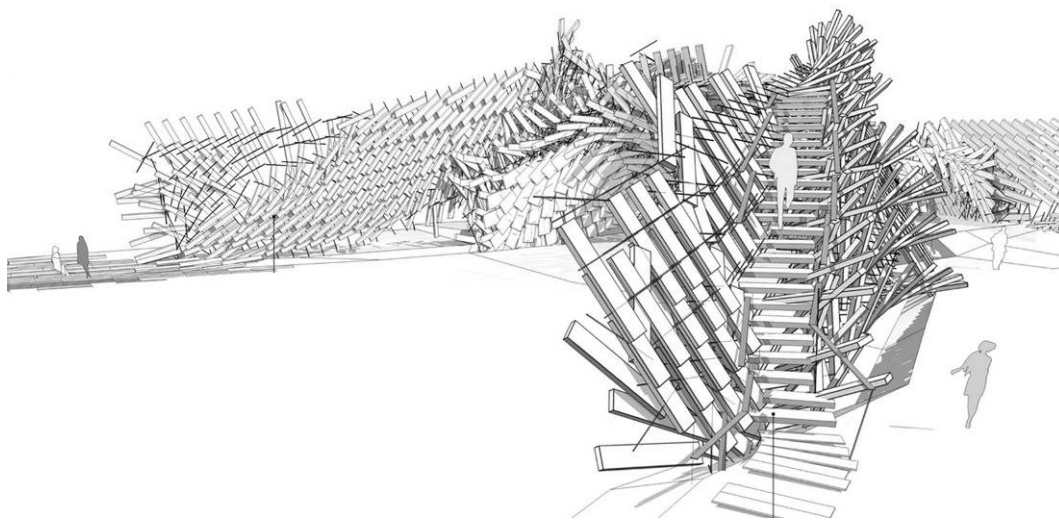


Figure 73.c : Instanciation des calepinages sur les surfaces du pavillon

**Figure 73 : « Radical Reproduction » D. Kroenhert et A. Orlinsky, CrossOver Studio (Université des Arts Appliquées, Vienne)**

Ce cas montre une différenciation explicite (et revendiquée) de la conception du système paramétrique, en particulier avec l'usage des modules de powercopies, par rapport à la conception de son instanciation. Trois *unités de conception* opèrent ici dans la conception du modèle paramétrique:

- celle de la modélisation des powercopies des calepinages : cette *unité de conception* relève de l'activité de conception de modèles paramétriques,

- celle de la modélisation des surfaces du pavillon : cette *unité de conception* relève également de l'activité de conception de modèles paramétriques,

- et celle de l'instanciation des powercopies de calepinage sur les surfaces : cette *unité de conception* relève de l'activité de conception de l'instance.

Il est intéressant de noter que les *espaces de référence* dont relèvent ici ces deux *unités de conception* ne sont pas les mêmes. Dans le premier temps, la conception du modèle paramétrique est principalement dominée par des pertinences géométriques et techniques liées aux performances mécaniques des assemblages de tasseaux de bois et aux géométries des surfaces. Dans le deuxième

temps, la conception de l'instance des calepinages sur les surfaces du « pavillon » fait appel à des *espaces de référence* plus variées, liés à l'échelle humaine, aux usages, aux vues créées sur le parc et à l'ensoleillement. Dans cet exemple, à chaque *unité de conception* du modèle paramétrique correspond donc plusieurs *niveaux de conception* différents.

La méta-modélisation opérant peu dans les cas analysés, nous nous focalisons dans cette thèse sur l'activité de conception du modèle paramétrique et de l'instance.

## 6.2.2 Propriétés d'échelles spécifiques à l'usage de la modélisation paramétrique

Lors de l'activité de conception de modèles paramétriques, il semble que certaines classes d'opérations de conception soient parfois dominantes. Une *opération de conception* est constituée à partir du domaine auquel appartient sa *pertinence* : c'est-à-dire du point de vue selon lequel est attribué la mesure. Ces points de vue peuvent être divers : relatif à la technique, au socioculturel, au symbolique, à la fonctionnalité de l'espace, *etc.* (Boudon et al. 2000, p.166-187). Dans l'usage de la modélisation paramétrique chez Gehry Partners et Foster and Partners (cf. Chapitre 3), il semble que certains points de vue soient récurrents lors de la conception de modèles paramétriques : les points de vue technique et optique chez Gehry Partners (cf. Chapitre 3.2) et, le point de vue technique chez Foster and Partners (cf. Chapitre 3.3).

En architecturologie, un point de vue est dit *dominant* lorsqu'il donne sa « coloration » à l'ensemble des opérations en jeu (Boudon et al. 2000, p.201). D'après Philippe Boudon, une classe de point de vue *dominant* peut caractériser un objet, une époque, *etc.* Il semble que les missions d'experts de la modélisation paramétrique puissent être caractérisées par une *dominance* de point de vues *techniques*, ce qui paraît conforme avec les missions qu'ils revendiquent (cf. Chapitre 3). Dans ce cas, l'*échelle technique* opère comme une *méta-échelle* dominante : toutes les *opérations de conception* sont concernées directement ou indirectement par cette classe. Cela ne signifie pas que l'*échelle technique* soit *principale*, c'est-à-dire qu'elle apparaisse le plus souvent (Boudon et al. 2000, p.201), puisque, nous l'avons vu (cf. Chapitre 3), de nombreux points de vue peuvent être impliqués dans la conception d'un modèle paramétrique. Mais lors de certains usages de la conception de modèles paramétriques par des experts, l'*échelle technique* est *dominante* et *structure* la majorité des opérations de conception impliquées. En architecturologie, une *échelle structurante* ordonne le système des points de vue d'un cas de conception (Boudon et al. 2000, p.201).

Par exemple, si l'on reprend un cas de la pratique de Gehry Partners, avec la conception du modèle paramétrique de calepinage du projet Fish (cf. 3.2.1.1.1), on observe la mise en œuvre de trois *échelles* :

- une *échelle optique*<sup>76</sup> (ou de *modèle*): le modèle paramétrique du calepinage prend en compte la façon dont l'objet sera vu pour correspondre aux intentions esthétiques du projet,

---

<sup>76</sup> L'*échelle optique* correspond à l'activité de « tenir compte d'un point de vue –au sens propre- pour donner à une partie ou à un tout de l'espace architectural une mesure qui fasse intervenir les modalités suivant lesquelles il est vu » (Boudon et al. 2000, p.167).

- une *échelle technique*<sup>77</sup> : la conception du modèle des surfaces constituant le calepinage prend en compte des considérations techniques liées à la construction de ces surfaces en bandes d'acier entremêlées,
- et une *échelle parcellaire*<sup>78</sup> : le modèle paramétrique du calepinage est conçu pour s'inscrire dans l'emprise de la surface de référence, modélisée à partir de la numérisation d'une maquette physique.

Dans la mise en œuvre de ces *échelles*, il semble que les *pertinences technique* et *optique* soient *structurantes*. Ainsi la pertinence de l'*échelle parcellaire* est induite par l'*échelle optique* (ou de *modèle*): s'il faut reprendre exactement l'emprise de la surface de référence, c'est pour que le modèle paramétrique de calepinage soit vu en rapport avec la géométrie définie par Frank Gehry.

On peut également prendre l'exemple du cas de la conception du modèle paramétrique de la façade du Walt Disney Concert Hall. Lors de la conception de ce modèle, les géométries ont été « rationalisées » par l'acteur de la conception de modèle (cf. 3.2.1.1.4) en vue de rendre l'objet architectural constructible avec les coûts et les techniques voulus. On observe ici la mise en œuvre des *échelles* :

- *parcellaire* : les nouvelles surfaces produites doivent correspondre aux surfaces de référence,
- *optique* : les nouvelles surfaces doivent pouvoir être vues comme les surfaces de référence,
- *technique* : les nouvelles surfaces doivent pouvoir être calepinées de façon à être construites facilement,
- *géométrique* : les solides euclidiens (tores, cylindres, etc.) sont pris comme référence pour la définition des nouvelles surfaces,
- *technique* : la rationalisation ou optimisation des surfaces vise à réduire le budget de construction du projet.

De nouveau, dans cet exemple les *échelles technique* et *optique* semblent *structurantes*. En effet, l'*échelle géométrique* est induite par l'*échelle technique* puisque la recherche de formes « rationnelles » (cf. 3.2.1.1.4) vise à rendre ces formes constructibles avec un calepinage de pierres. Et l'*échelle parcellaire*, comme dans le cas de la modélisation du calepinage du projet Fish, est induite par l'*échelle optique* : les nouvelles formes modélisées doivent pouvoir être vues comme les formes initialement conçues.

Pour le cas des missions de Gehry Technologies (de 2003 à aujourd'hui) et C-Cubbed (entre 1991 et 2003), la dominance des *échelles techniques* et *optique* expriment les enjeux de représentation et de fabrication, majeurs dans l'usage de la modélisation paramétrique des cas observés (cf. Chapitre 3).

---

<sup>77</sup> L'*échelle technique* correspond à l'activité d'« Utiliser des considérations d'ordre technique pour induire une modalité d'attribution de mesures à une partie ou un tout de l'espace architectural » (Boudon et al. 2000, p.175).

<sup>78</sup> L'*échelle parcellaire* correspond à l'activité de « Mettre en œuvre des mesures en utilisant les possibles permis par la taille, la forme et les limites du terrain donné à l'architecte. » (Boudon et al. 2000, p.172).

### 6.3 Relations entre opérations de conception du modèle paramétrique et opérations de conception architecturale

La relation entre *dimension* du modèle paramétrique et *dimension* de l'objet architectural pourrait être interrogée en termes d'assistance. L'assistance à la conception architecturale par la modélisation paramétrique n'est pas l'objet de recherche de cette thèse. L'objet de la recherche est de construire une connaissance des opérations cognitives impliquées dans la conception de modèles paramétriques en conception architecturale. Cette section de la thèse interroge les relations entre opérations de conception de modèles paramétriques et opérations de conception architecturale (cf. Figure 74).

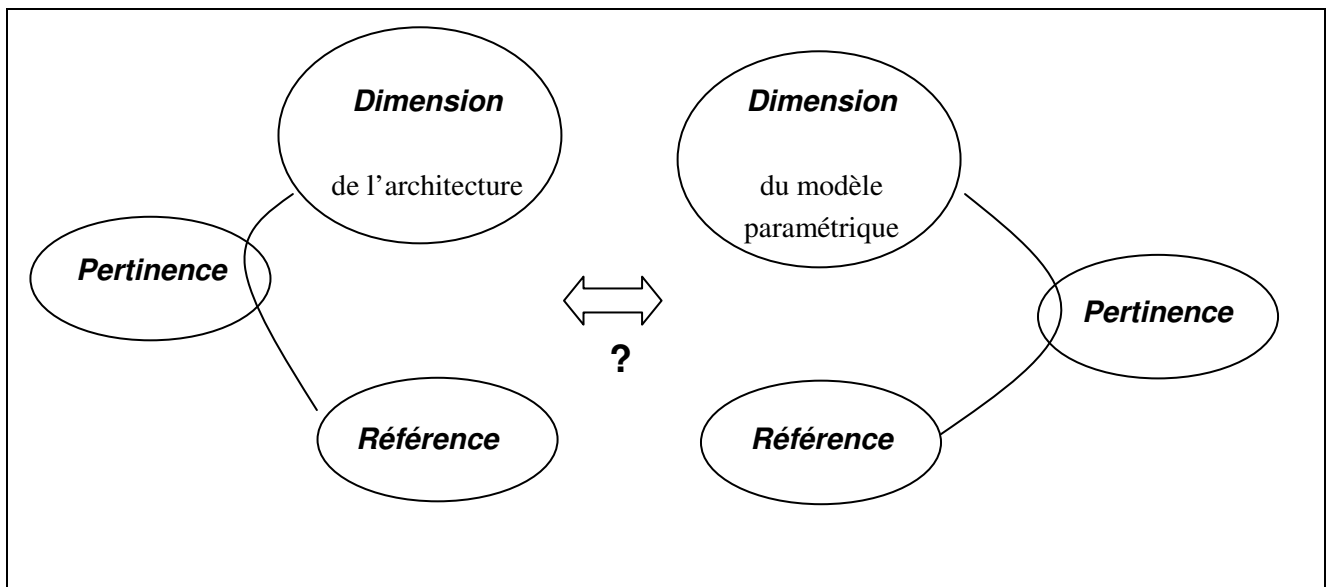


Figure 74: Vers une schématisation des relations d'opérations entre conception de modèles paramétriques et conception architecturale

L'outillage théorique mis en place dans cette recherche distingue les *opérations de conception de modèles paramétriques* et les *opérations de conception architecturale*. Cette distinction permet d'interroger les mécanismes cognitifs liant les activités de conception architecturale et de modélisation paramétrique en termes de relations d'opérations. En architecturologie, des concepts permettent d'interroger les relations qu'entretiennent des opérations de la conception (Boudon et al. 2000, p.197). Ceux-ci interrogent :

- les relations temporelles des opérations de conception: ce sont les relations en *relais* ou en *cascade*, elles seront abordées dans le paragraphe 6.3.1 (cf. p. 204),
- les relations spatiales des opérations de conception: ce sont les relations de *surdétermination*, de *codétermination* ou de *juxtaposition*, elles seront abordées dans le paragraphe 6.3.2 (cf. p. 208).



### 6.3.1 Relations temporelles : opérations en *relais* ou *cascade*

L'architecturologie propose les concepts de *relais* et de *cascade* (Boudon et al. 2000, p.238) pour interroger les relations qu'entretiennent plusieurs opérations de conception portant sur un même support de conception (une *dimension*) mais distinctes dans le temps (une opération précédant ou suivant l'autre). Ces relations entre opérations sont pensées au sein d'un même *espace de la conception*. Un *espace de conception* est un espace théorique propre à un processus de conception singulier, empiriquement « observable » (Lecourtois 2006a, p.86). Les *opérations de conception de modèles paramétriques* et les *opérations de conception architecturale* sont dans l'*espace de la conception*, mais sont, en théorie, spécifiques à des *espaces de conception* différents : l'*espace de conception* du projet architectural d'un côté et l'*espace de conception* du modèle paramétrique de l'autre (cf. Chapitre 5). Nous proposons, dans cette recherche, d'étendre les concepts de *relais* et de *cascade*, en vue d'interroger les relations entre des opérations inscrites dans des espaces de conception distincts.

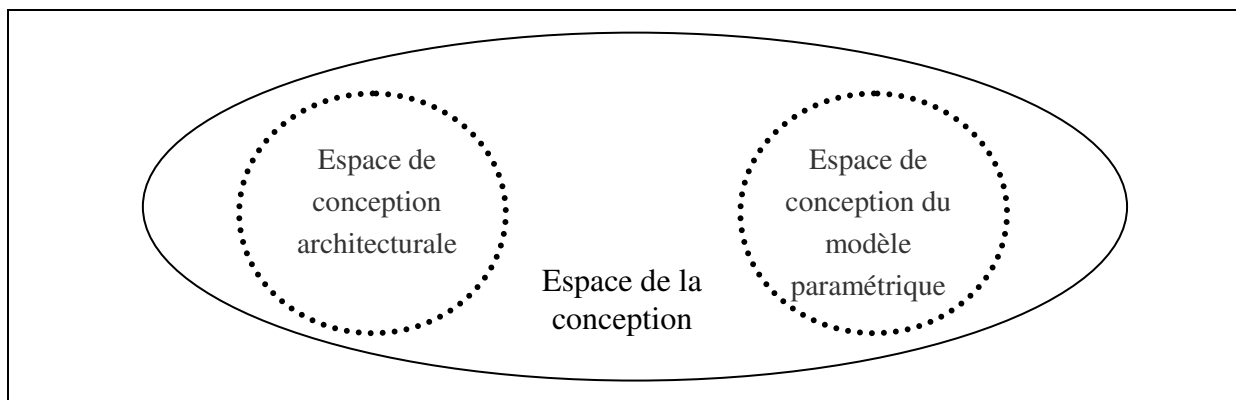
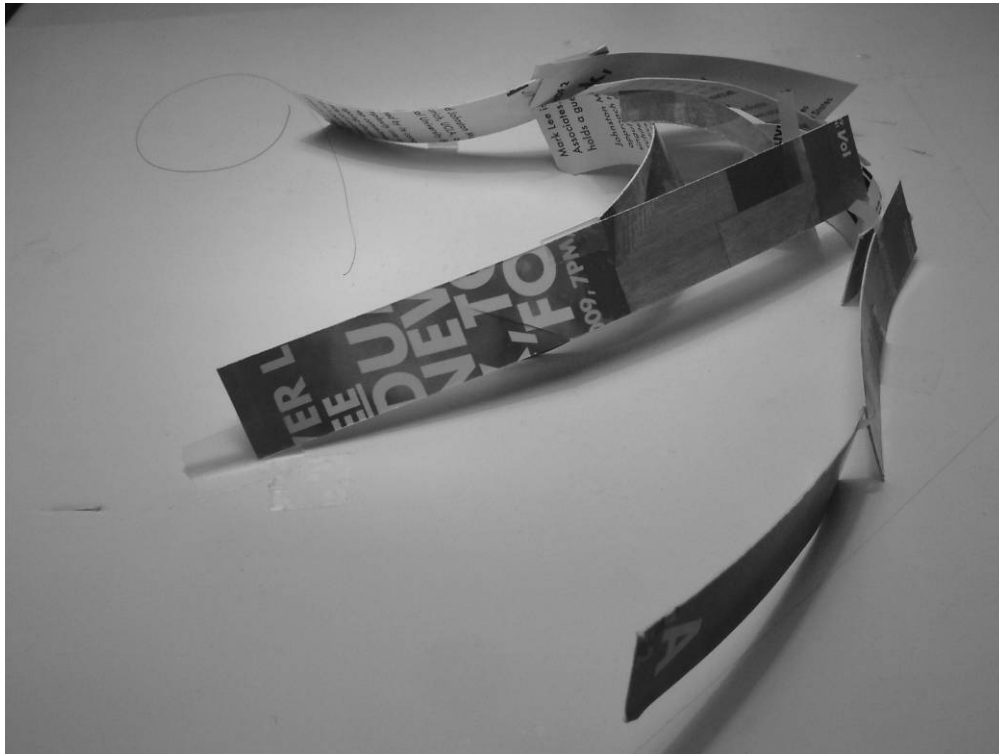


Figure 75 : Espaces de conception / espaces de la conception

Le *relais* désigne une opération suivie d'une autre sans nécessité : chaque opération relève d'une intentionnalité spécifique. Dans une relation en *cascade* les opérations se suivent nécessairement, une opération en produisant ou induisant une autre bien déterminée.

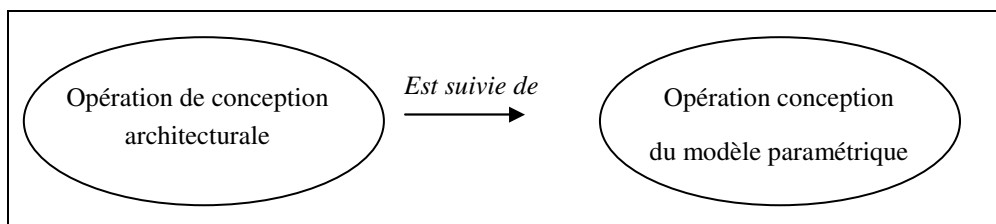
Dans les cas analysés, il est difficile d'évaluer le caractère nécessaire entre les opérations. Nous ne présenterons donc ici que des exemples de *relais*, c'est-à-dire des cas où une décision relative à une *pertinence* motive la mise en œuvre d'une nouvelle *pertinence* choisie librement.



**Figure 76 : « Radical Production », maquette d'étude, source: Adam Orlinsky**

Dans le projet « Radical Production » (cf. 5.5.1), les géométries des surfaces et leur positionnement dans le site ont été pensés à partir de maquettes physiques (cf. Figure 76) et relativement à l'usage du pavillon (*échelle fonctionnelle*), à la proximité de monuments sur le site (*échelle de voisinage*), et à la volonté de créer des vues spécifiques sur la ville (*échelle de visibilité*). Le modèle paramétrique a ensuite été conçu pour représenter les surfaces en fonction de ces *pertinences*.

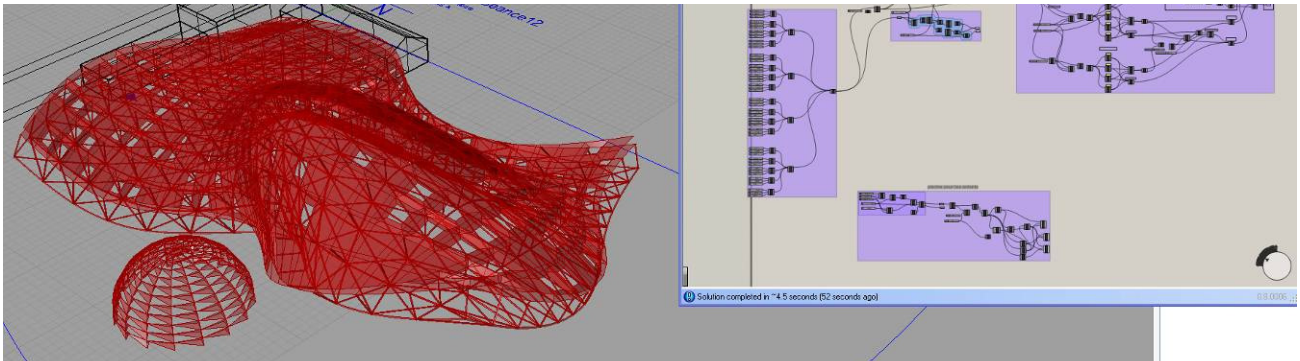
La conception du modèle paramétrique des surfaces de ce projet intervient donc ici en *relais* après une opération de conception architecturale (cf. Figure 77). Cette relation peut-être suivie d'autres opérations de conception.



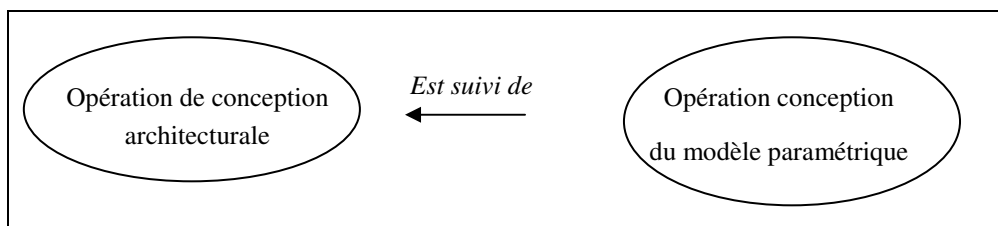
**Figure 77: Relais de la conception du modèle paramétrique après la conception architecturale**

Dans d'autres exemples, on observe des cas de *relais* d'opérations entre conception de modèles paramétriques et conception architecturale. C'est le cas dans le projet de piscine mené par Toshihiro Kubota et Pierre Rousteau à l'ENSA de Paris la Villette, dans le cadre de l'Expe1. Lors de la conception du projet, les étudiants ont exploré les différentes possibilités à leur disposition pour calepiner rapidement une surface sur Grasshopper. Le choix de deux calepinages à utiliser dans la modélisation paramétrique a amené les concepteurs à proposer deux bassins pouvant intégrer des couvertures, chacune calepinée différemment (cf. Figure 78). La librairie de modèles paramétriques de calepinages a

servi de support d'inspiration à la modélisation paramétrique. La conception des modèles de cette librairie a précédé dans le temps l'opération de conception architecturale de ce projet visant à choisir un calepinage. Dans cet exemple, l'opération de conception architecturale succède à l'opération de conception du modèle paramétrique (cf. Figure 79).

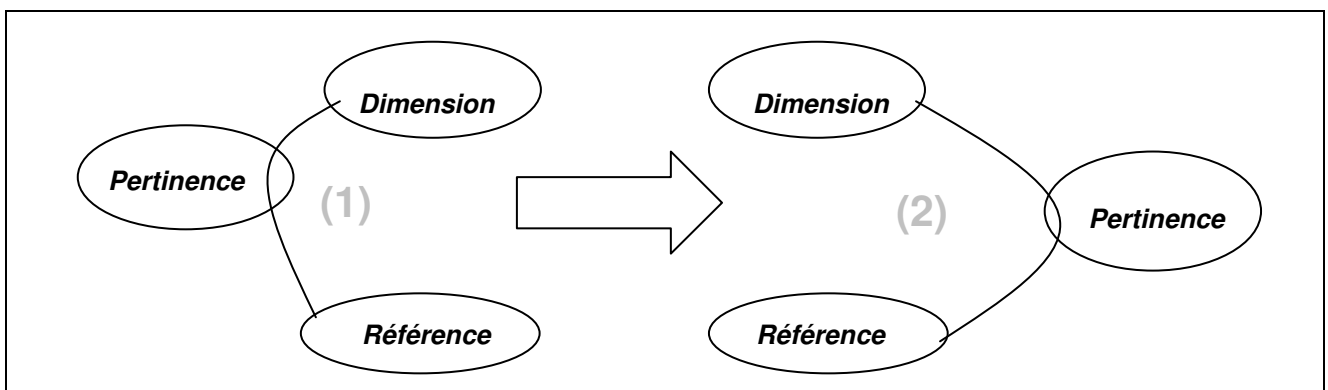


**Figure 78 : Projet de piscine, par Toshihiro Kubota et Pierre Rousteau, sources : A. de Boissieu d'après le modèle de Kubota et Rousteau**



**Figure 79: Inférence de la modélisation paramétrique vers la conception architecturale**

Cet exemple pointe un cas récurrent qui sera tout particulièrement étudié dans le chapitre suivant : celui de la réutilisation de modèles paramétriques existants. Dans ces cas là, des opérations de conception architecturale succèdent aux opérations de conception de modèles paramétriques ayant donné lieu aux modèles réutilisés. De nouvelles opérations, à la fois de conception architecturale et de conception du modèle, opèrent ensuite pour poursuivre la conception du projet en allers-et-retours avec le modèle découvert ainsi que pour modifier le modèle et l'adapter aux intentions du projet.



**Figure 80: Schématisation de la relation « en relais » entre opérations de conception**

La distinction dans le temps de la conception de modèles paramétriques et de la conception architecturale entraîne des *relais* opérant potentiellement de l'une vers l'autre (cf. Figure 80). Les opérations de conception architecturale et de conception de modèles paramétriques sont donc potentiellement prises dans une chaîne. Les passages d'une opération à l'autre sont plus particulièrement interrogés dans la suite du chapitre à travers la possibilité d'interventions d'opérations tierces (cf. section 6.4).

### 6.3.2 Relation dans l'espace : juxtaposition, partage de référence et/ou de pertinence

Les relations temporelles ne sont pas les seules à pouvoir décrire les liens entre plusieurs opérations. Les opérations de conception architecturale et les opérations de conception de modèles paramétriques sont-elles totalement hermétiques les unes aux autres ? Entretiennent-elles des relations spatiales ?

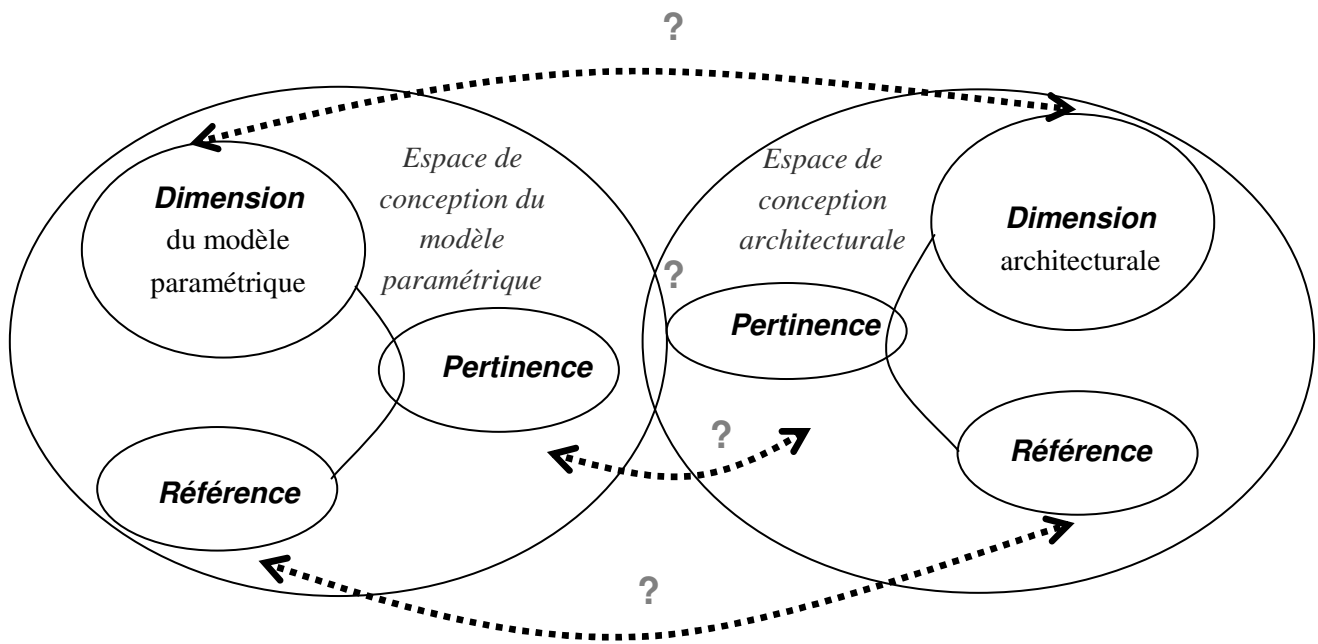
En architecturologie, plusieurs concepts permettent d'interroger les relations entretenues par des opérations portant sur des *dimensions* situées les unes par rapport aux autres. Ces concepts sont : les relations de *surdétermination*, de *juxtaposition* et de *codétermination* (Boudon et al. 2000, p.198).

Dans une relation dite de *surdétermination*, les opérations portent sur une même *dimension*. Ces opérations de conception restent indépendantes : elles se surajoutent simplement pour donner de la mesure à un même objet. Dans le cas de notre recherche, on ne peut pas assimiler une *dimension* du modèle paramétrique à une *dimension* architecturale, même si celles-ci sont souvent très proches.

La *juxtaposition* quant à elle désigne la relation entretenue par plusieurs opérations qui coexistent en portant sur des *dimensions* différentes. Comme dans le cas des relations en relais ou en cascade vues précédemment, dans la littérature architecturologique ce concept désigne des relations d'opérations appartenant à un même *espace de conception* (Boudon et al. 2000, p.198). Nous devons donc ici encore étendre la définition de relation de *juxtaposition* pour pouvoir interroger les relations qu'entretiennent opérations de conception architecturale et opérations de conception de modèle paramétrique.

Le concept de *juxtaposition* d'opérations s'oppose à la *surdétermination* et désigne des mesures qui « coexistent » sans avoir le même support. Or, on peut s'interroger ici sur certaines modalités de coexistence d'opérations de conception de modèles paramétriques et d'opérations de conception architecturale. Si ces opérations portent sur des *dimensions* différentes (l'une porte sur le modèle paramétrique, l'autre sur l'objet architectural), sont-elles totalement hermétiques l'une à l'autre ? N'entretiennent-elles que des relations temporelles de succession ?

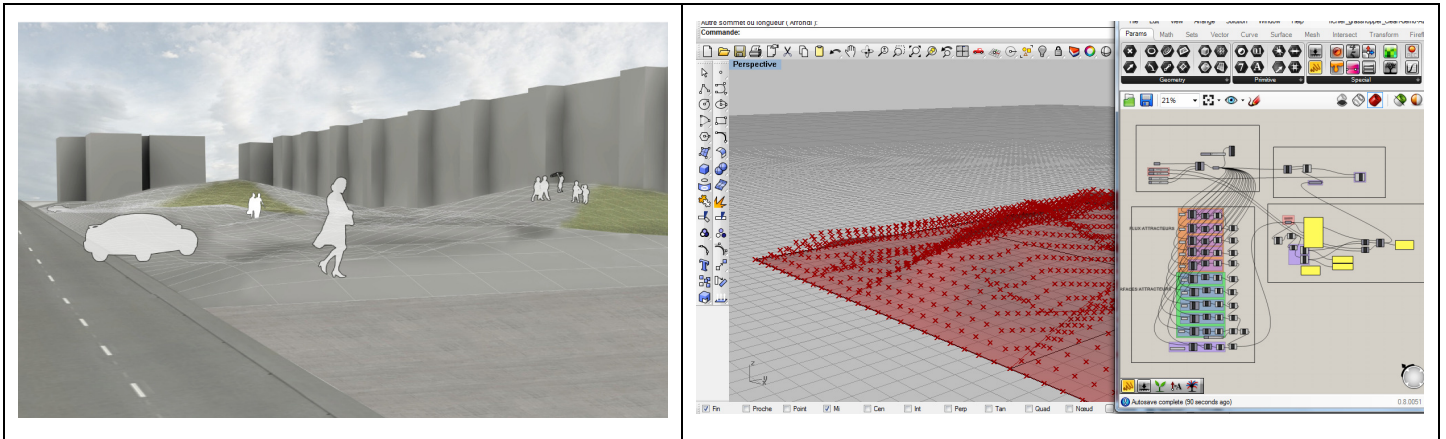
Dès l'interrogation de possibles *opérations élémentaires* de conception de modèles paramétrique (cf. 6.1), nous avons relevé des possibilités d'entremêlement entre conception architecturale et conception de modèle paramétrique. Malgré notre distinction théorique entre *espace de conception* du modèle paramétrique et *espace de conception* architecturale, il semble que des croisements opèrent parfois (cf. Figure 81). Quels sont-ils ?



**Figure 81: Dans une relation de juxtaposition, quels sont les croisements possibles entre opérations de conception du modèle paramétrique et opération de conception architecturale?**

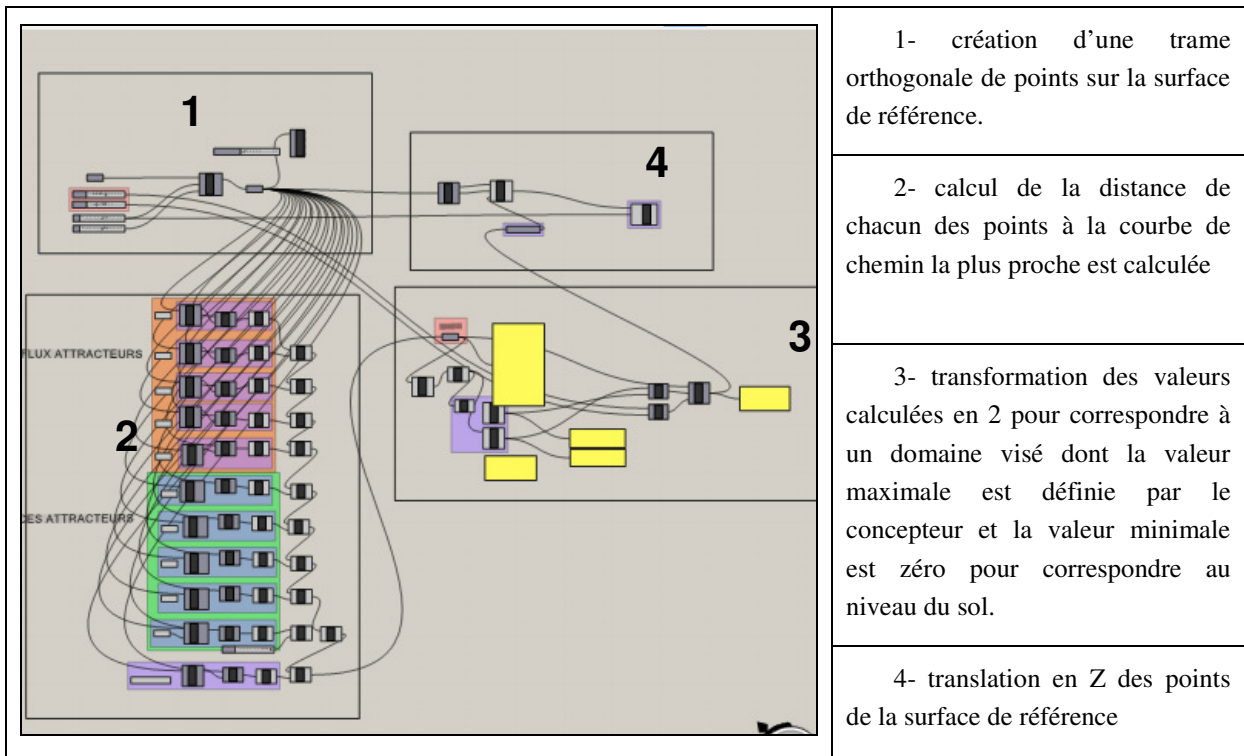
Ces croisements possibles entre opérations, malgré des *dimensions* et des *espaces de conception* distincts, semblent pouvoir être des croisements par partage de *pertinences* ou bien des croisements par partage d'*espaces de référence*. Nous utilisons ici « partage » dans le sens courant d'« avoir part à quelque chose en même temps que d'autres » (Rey-Debove & Rey 1992, p.1364).

On observe un cas de partage de *pertinences* et d'*espaces de référence* dans le processus du projet « Topographie » (cf. Figure 82), abordé dans le paragraphe 6.1.2. Dans ce projet, des opérations de conception portant sur des *dimensions* distinctes peuvent être identifiées. Ces *dimensions* sont : la *dimension* de la topographie du projet, c'est-à-dire la surface conçue par le projet (cf. Figure 82 à gauche) et la *dimension* de la surface du modèle paramétrique (cf. Figure 82 à droite). Nous l'avons vu en 6.1.2, ces opérations sont distinctes mais s'entremêlent toutefois. Il semble qu'elles partagent des *pertinences* et des *espaces de référence* et nous allons voir lesquels.



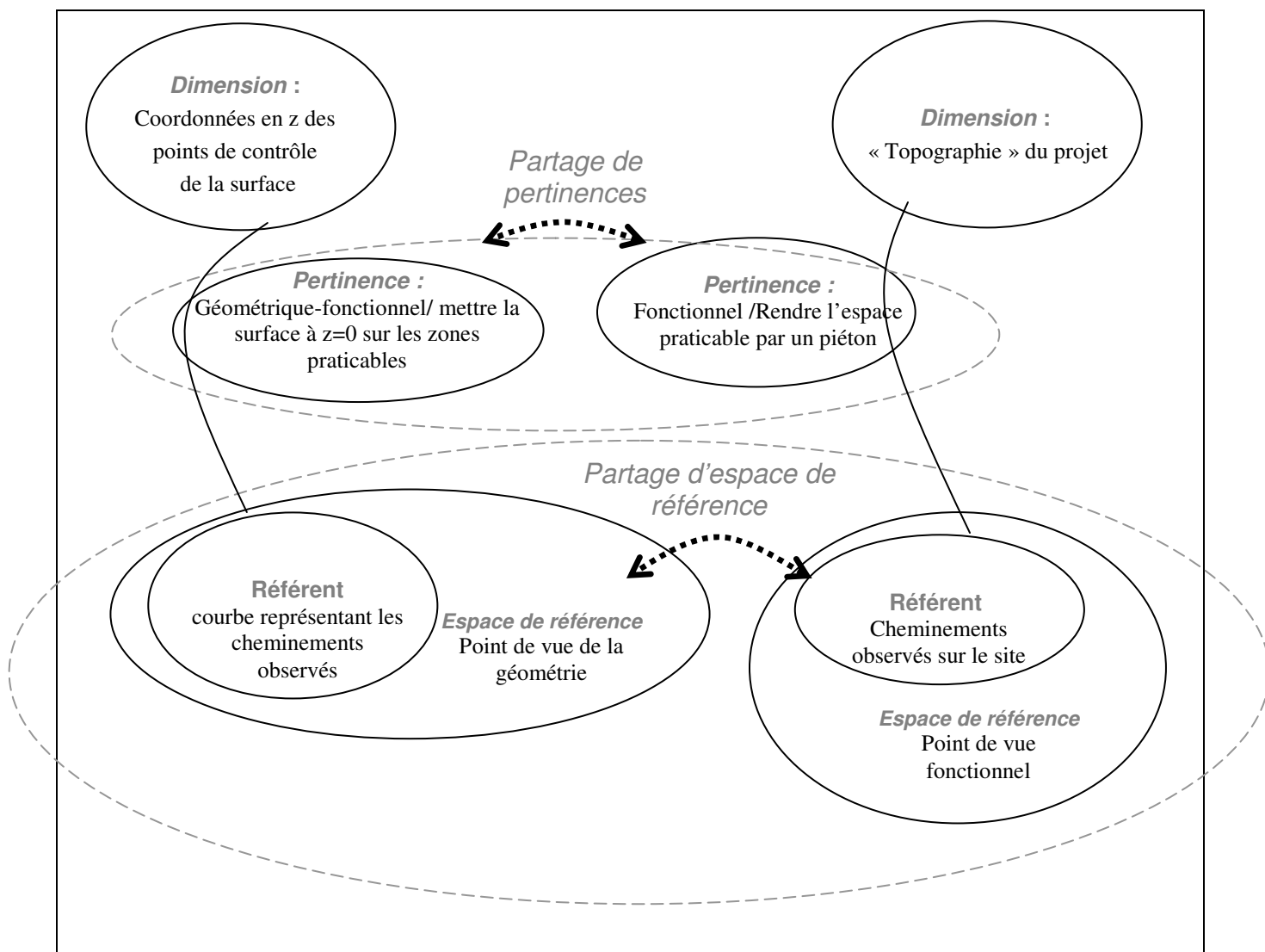
**Figure 82: Projet Topographie, à gauche la « surface » comme *dimension* architecturale et à droite la « surface » comme *dimension* du modèle paramétrique**

Dans ce cas, les concepteurs ont conçu la *dimension* « surface » du modèle paramétrique à partir des *pertinences* mises en œuvre pour la *dimension* « topographie » de l'objet architectural. La *pertinence* utilisée pour *dimensionner* la topographie de l'objet architectural est de rendre certains des espaces praticables pour les piétons. Cette *pertinence* est également mise en œuvre, sous la forme d'une interprétation pour concevoir la surface du modèle paramétrique et, en particulier, pour définir les altitudes des différents points de la surface. En effet, les coordonnées en Z des points de la surface du modèle paramétrique dépendent de la distance de ces points à des courbes représentant les principaux chemins observés sur le site (cf. Figure 83). Plus un point est proche d'une zone de circulation, plus son altitude est faible.



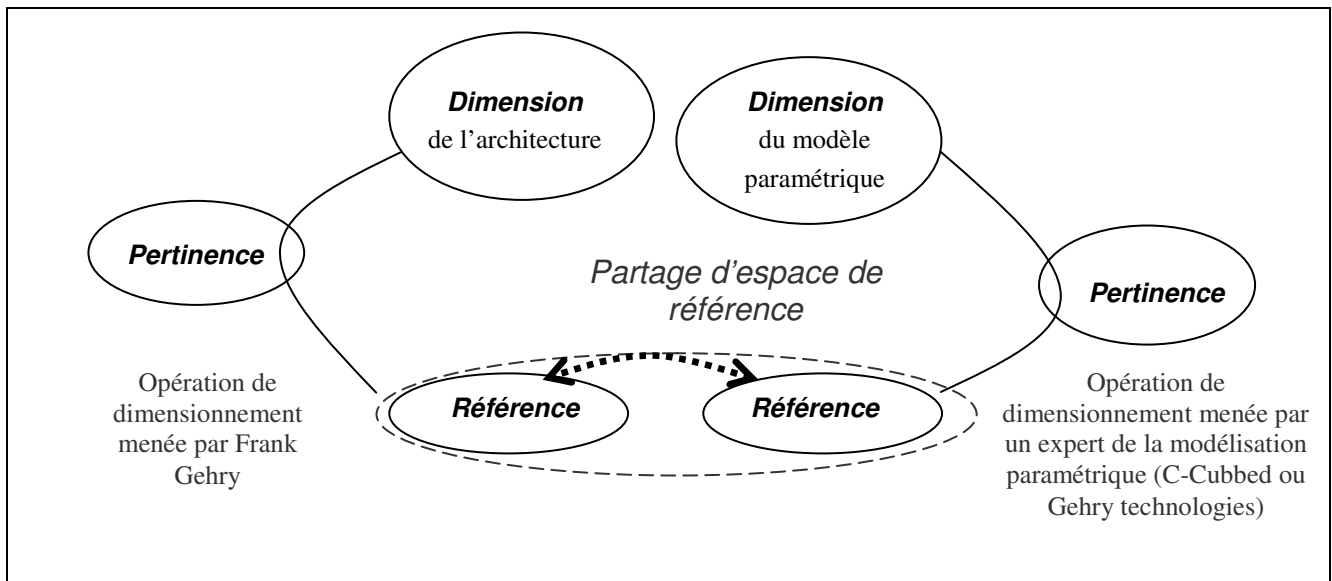
**Figure 83 : Image du modèle paramétrique du projet « Topographie » et description cf. 6.1.2**

Du point de vue architecturologique, tout se passe comme si *pertinence* et *espace de référence* étaient partagés, non pas « tels quels », mais par une adaptation ou *interprétation* de ceux-ci d'un *espace de conception* à l'autre (cf. Figure 84). Nous interrogerons cette idée « d'interprétation » dans la suite du texte.



**Figure 84: Exemple d'opérations avec partage de pertinence et de référence dans le cas du projet « Topographie »**

On observe un autre exemple de partage d'*espaces de référence* dans le cas de la conception de Frank Owen Gehry, tel que décrit précédemment (cf. 3.2.1.2). Après 1991, Frank Gehry conçoit des architectures avec un imaginaire relatif aux potentialités techniques de la modélisation paramétrique. Sans manipuler ou savoir concevoir un modèle paramétrique, Frank Gehry comprend les potentialités de la modélisation paramétrique. Cela constitue un *espace de référence* permettant à Frank Gehry de concevoir en prenant en compte de possibles futures *opérations de conception du modèle paramétrique*. Dans ce cas, conception architecturale et conception du modèle paramétrique sont menées par des acteurs différents, mais ces activités sollicitent des *espaces de référence* qui renvoient l'un à l'autre (cf. Figure 85).



**Figure 85: Partage d'un espace de référence entre opérations de conception architecturale menées par Frank Gehry et opérations de conception du modèle paramétrique menées par C-Cubbed ou Gehry Technologies**

Comment s'opèrent ces partages de *pertinences* et de *références* ? Nous interrogeons dans la suite de ce chapitre la possibilité d'intervention d'opérations tierces, ne relevant pas directement de l'opération de conception mais intervenant cependant sur la conception.

## 6.4 Opérations tierces impliquées

Les premiers éléments de connaissance, développés dans ce chapitre sur la conception de modèles paramétriques en conception architecturale, pointent l'occurrence d'opérations autres que des attributions de mesures.

Ces opérations tierces semblent se rapprocher de ce que Samia Ben Rajeb nomme « opérations pragmatiques ». Le concept « d'opération pragmatique » est proposé par Samia Ben Rajeb dans sa thèse *Modélisation de la collaboration distante dans les pratiques de conception architecturale : Caractérisation des opérations cognitives en conception collaborative instrumentée* (Ben Rajeb 2012, p.281). Cette recherche définit les *opérations pragmatiques* comme des opérations cognitives de conception d'une situation de conception. C'est-à-dire comme des opérations participant à la conception sans directement être des opérations d'attribution de mesures (Ben Rajeb 2012, p.281). Dans le cadre de sa recherche, Ben Rajeb s'est particulièrement intéressée aux opérations pragmatiques permettant de construire un « espace de conception de la situation de conception architecturale », en particulier dans un cadre de collaboration outillée synchrone à distance (Ben Rajeb 2012, p.271). Dans ce travail, Ben Rajeb distingue des *opérations pragmatiques* de collaboration et d'usage de l'outil (*cf.* Tableau 30).



| <i>Opérations élémentaires de conception</i> | <i>Opérations pragmatiques</i>    |                                |
|--|-----------------------------------|--------------------------------|
|  | <i>de collaboration</i>           | <i>d'usage de l'outil</i>      |
|  | Prescription                      |                                |
|  | Autonomisation                    |                                |
|  | Interprétation                    |                                |
|  | Mise en commun                    |                                |
|  | Evaluation                        | Evaluation                     |
|  | Construction de règles de travail | Construction de règles d'usage |
| Découpage                                    | Découpage (structuration)         | Découpage (structuration)      |
| Référenciation                               | Référenciation                    | Référenciation                 |
| Dimensionnement                              |                                   | Dimensionnement                |
|  |                                   | Appropriation                  |
| Remise en cause                              |                                   |                                |

**Tableau 30 : Liste des différentes opérations cognitives mises en jeu dans le cadre de la collaboration distante outillée en conception architecturale (Ben Rajeb 2012, p.276)**

Dans le cadre de notre recherche sur la conception de modèles paramétriques en conception architecturale, nous reprendrons plus particulièrement les notions d'opérations *d'interprétation*, *de mise en commun* et *d'appropriation de l'outil*.

Dans le cadre de la conception de modèles paramétriques en conception architecturale, trois sortes d'opérations pragmatiques semblent opérer :

- des opérations de collaboration (6.4.1) : l'opération *d'interprétation* et l'opération de *mise en commun*;
- des opérations d'usage de l'outil (6.4.2) : l'opération *d'interprétation de l'outil* et l'opération *de traduction en géométrie paramétrique* ;
- des opérations logiques (6.4.3) : l'opération *d'induction* et l'opération de *vérification*.

## 6.4.1 Opération pragmatique de collaboration

### 6.4.1.1 Opération d'interprétation

L'*opération pragmatique d'interprétation* spécifique à la collaboration est une opération par laquelle un concepteur interprète ce que dit ou dessine son collaborateur et par laquelle il se construit de nouveaux points de vue (Ben Rajeb 2012, p.285). Ces points de vue sont partagés par différents acteurs. Le terme « Interpréter » est ici pris dans son acception courante, telle que définie par le Centre National de Ressources Textuelles et Lexicales : « *Action de donner un sens personnel, parmi d'autres possibles, à un acte, à un fait, dont l'explication n'apparaît pas de manière évidente* » [cnrtl.fr/definition/]. Par l'interprétation des propos de son (ses) interlocuteur(s), un concepteur explicite des informations et se les approprient en leur donnant une signification personnelle. L'interprétation peut être l'objet d'une co-construction, permettant à plusieurs collaborateurs de se construire des références communes (Ben

Rajeb 2012, p.285). Cette *opération pragmatique* rend possible le partage de points de vue pour des opérations de conception différentes (cf. Figure 86).

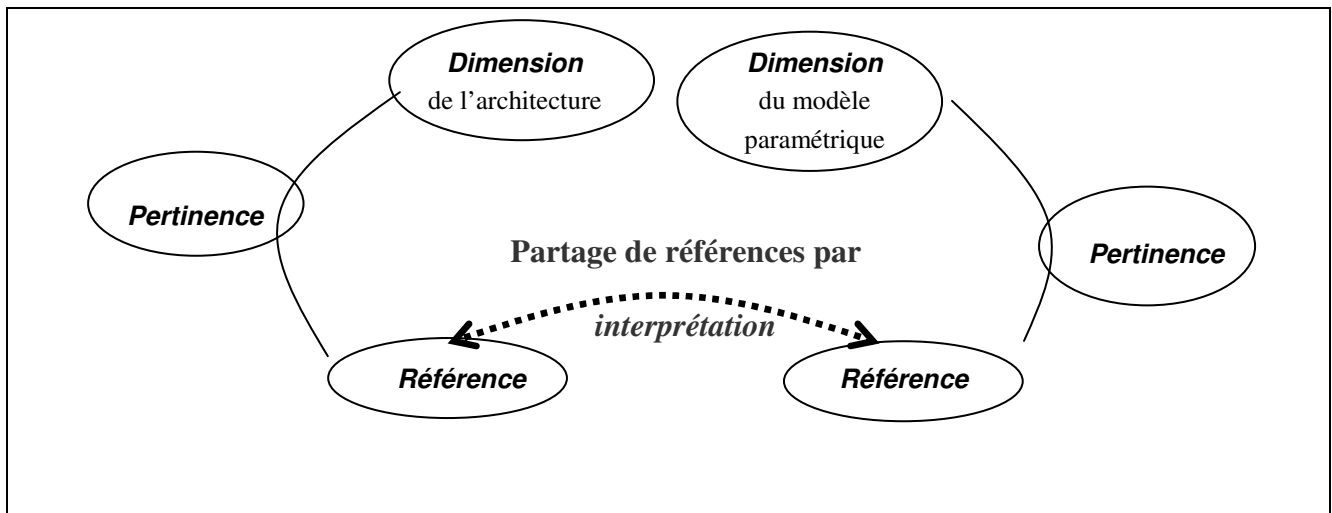


Figure 86: *Opération d'interprétation pour la collaboration, permettant un partage de point de vue*

Les nouveaux savoirs construits par l'*interprétation* permettent aux concepteurs de se construire de nouveaux points de vue pouvant générer de nouvelles *pertinences*.

Nous avons pu observer dans le Chapitre 3 que l'usage de la modélisation paramétrique en agence d'architecture était, le plus souvent, le fait d'experts distincts des acteurs de la conception architecturale. Dans ces cas, certains experts affirment qu'une part importante de leur travail est l'interprétation des intentions architecturales des équipes de conception (Whitehead 2009b; Kocaturk & Medjdoub 2011). Dans le cas des activités du SMG et de l'ARD, cette interprétation des désirs et des intentions architecturales des équipes de conception est un support pour la conception de modèles paramétriques (cf. 3.3). Il semble qu'une certaine forme d'interprétation soit indispensable pour satisfaire les intentions des concepteurs. Cette même interprétation mène à la construction de référence et de *pertinence* partagées pour la conception de modèles paramétriques.

L'*interprétation* par et pour la collaboration opère pour l'acteur de la modélisation paramétrique qui reçoit la formulation de l'état ou des désirs de conception architecturale. Mais, l'opération d'interprétation pour la collaboration opère également pour l'acteur de la conception architecturale qui traduit ses propres idées dans cet échange avec l'acteur de la modélisation paramétrique. Ainsi, l'acteur de la conception architecturale est susceptible de revoir son travail du point de vue d'une possible conception de modèle paramétrique à partir de celui-ci. Pour Christophe Camus, la description d'objets architecturaux peut opérer comme une « *négociation* » que le récit descriptif permet de « *créer ou mettre en forme, en l'unifiant ou en le stabilisant architecturalement* » (Camus 1996, p.239). Pour Camus, cette pratique de la narration fait partie intégrante du processus de la conception architecturale : la situation d'énonciation de l'architecture peut être structurante pour la conception en particulier par le développement d'une compétence de la traduction intersémiotique (Camus 1996, p.242)<sup>79</sup>. Énoncer son projet d'architecture en vue de le faire comprendre à un collaborateur implique l'interprétation de son

<sup>79</sup> « Et il semble qu'on peut trouver, dans l'activité de description architecturale, quelque chose de l'ordre de l'exercice, permettant non seulement de développer des compétences communicationnelles, mais aussi une compétence créative qui s'inscrit dans une culture de la traduction intersémiotique pratiquée dès le travail de conception. » (Camus 1996, p.242).

propre travail sous un nouveau regard, ici celui de la construction de modèles paramétriques. Par la narration et l'interprétation mise en œuvre dans ces situations d'échanges entre acteurs, *pertinences*, *espaces de référence* et mesures sont restructurés. L'acteur de la conception et l'acteur de la modélisation paramétrique sont susceptibles de faire évoluer et d'entremêler les mesures, *pertinences* et références mises en œuvre.

L'*opération pragmatique d'interprétation* peut également opérer dans le cas où un architecte, non expert en modélisation paramétrique, s'intéresse à la conception d'un modèle paramétrique et cherche des ressources pour mener à bien ce modèle. Ces ressources, construites par d'autres, sont interprétées par le concepteur en vue de devenir opérantes pour les opérations mises en œuvre.

#### 6.4.1.2 Opération de mise en commun

L'opération de *mise en commun* est définie par Samia Ben Rajeb comme une opération de construction de la situation de conception architecturale. C'est une opération par laquelle des concepteurs échangent des informations en vue de se construire un *référentiel opératif commun*, pouvant par la suite être sollicité pour la conception (Ben Rajeb 2012, p.286)<sup>80</sup>.

Dans la littérature architecturologique, un *espace de référence* est une construction mentale éminemment personnelle (Boudon et al. 2000, p.110). Un *référentiel opératif commun*, quant à lui, est un point de vue construit et partagé par plusieurs acteurs et est indispensable à la prise de décision à plusieurs (de Terssac & Chabaud 1990), pour plusieurs raisons : la compréhension des points de vues mutuels, le partage d'informations communes, etc. Par ailleurs, selon Terssac et Chabaud (1990), le *référentiel opératif commun* permet également la mise en commun de compétences spécifiques.

Par l'opération *d'interprétation*, acteurs de la conception architecturale et experts de la modélisation paramétrique se construisent des *espaces de référence* personnels à partir des propos qu'ils peuvent échanger. Ces nouveaux *espaces de référence* leur permettent d'intégrer des savoirs autres dans la mise en œuvre de leurs propres compétences.

Par l'opération de *mise en commun*, en revanche, acteurs de la modélisation paramétrique et acteurs de la conception architecturale se construisent un point de vue commun leur permettant de partager des compétences.

---

<sup>80</sup> Les opérations de mises en commun « permettent d'ajouter et d'apporter des informations qui paraissent indispensables à la collaboration et qui concernent directement ou indirectement l'objet à concevoir. Ces mises en commun sont donc les opérations pragmatiques par lesquelles les concepteurs se construisent, graphiquement et/ou oralement, un référentiel opératif commun au fur à mesure de l'évolution du processus de conception. » (Ben Rajeb 2012, p.286).

## 6.4.2 Opérations pragmatiques d'usage de l'outil

### 6.4.2.1 Opération d' appropriation de l'outil

L'opération pragmatique d'appropriation de l'outil semble également intervenir lors d'usage de la modélisation paramétrique en conception architecturale. L'opération pragmatique d'appropriation de l'outil est l'opération par laquelle « le(s) concepteur(s) *interprète(nt)* et *s'approprie(nt)* l'outil, son interface et ses fonctionnalités pour répondre aux besoins du projet (la modélisation d'une pensée cognitive, la simulation de l'objet en train d'être conçu et/ou la communication de ses choix à l'autre). » (Ben Rajeb 2012, p.310). L'opération pragmatique d'appropriation de l'outil peut prendre des formes diverses (Ben Rajeb 2012, p.310) :

-« l'appropriation de l'outil relativement aux besoins du projet » : cela concerne par exemple le choix d'un modèleur spécifique en fonction d'un projet. Pour la modélisation paramétrique, cette opération n'est pas anodine car les modèleurs paramétriques ne sont encore que rarement utilisés par les architectes (cf. Chapitre 3). Dans le cas de la modélisation paramétrique, cette appropriation permet aux concepteurs de se construire un *espace de référence* sur les spécificités des modèleurs qu'ils connaissent, à partir duquel sera sollicité un modèleur ou une fonctionnalité spécifique pour la mise en œuvre d'opérations de conception.

-« le choix des fonctions » : cette opération est relative à l'interprétation et l'utilisation d'une fonction par un concepteur. Par exemple, dans le cas du projet « BioCity » (cf. annexes p.367) la conception du modèle paramétrique a pris en compte les performances de l'outil : les modèles ont été conçus en vue d'optimiser les temps de calcul.

-« les interactions avec l'outil » : cette opération rend compte du « degré d'acceptation et/ou de consentement pour l'appropriation de l'outil et de ses fonctionnalités » (Ben Rajeb 2012, p.310). Par exemple pour certains projets de l'atelier CrossOver, de nombreux abandons de Digital Project ont été observés : ce modèleur a été considéré comme trop complexe et trop lourd pour que la conception de modèles paramétriques soit poursuivie par ces acteurs.

Dans le cas de la conception de modèles paramétriques pour la conception architecturale, on observe que l'opération d'*appropriation* de l'outil est en fait une interprétation propre au concepteur des spécificités des modèleurs. Par celle-ci, le concepteur construit un univers de référence sur les fonctionnalités des modèleurs, leurs mécanismes de propagation, leurs difficultés d'usage ou encore les représentations qu'ils facilitent. Ces *espaces de référence* peuvent être partagés entre opérations de conception du modèle paramétrique et opérations de la conception architecturale (cf. Figure 86).

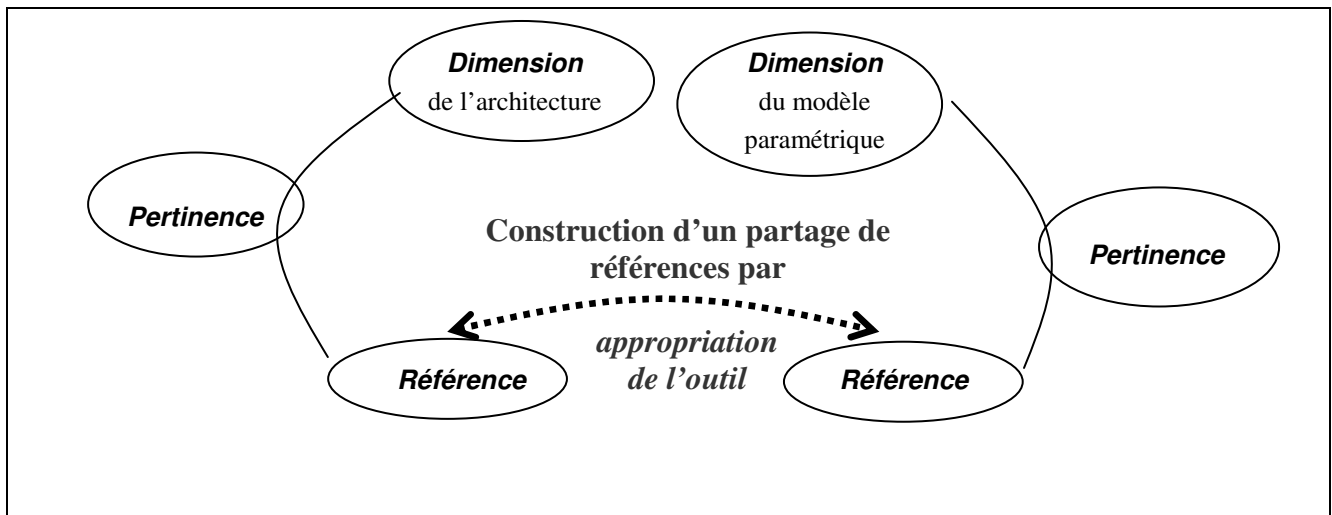


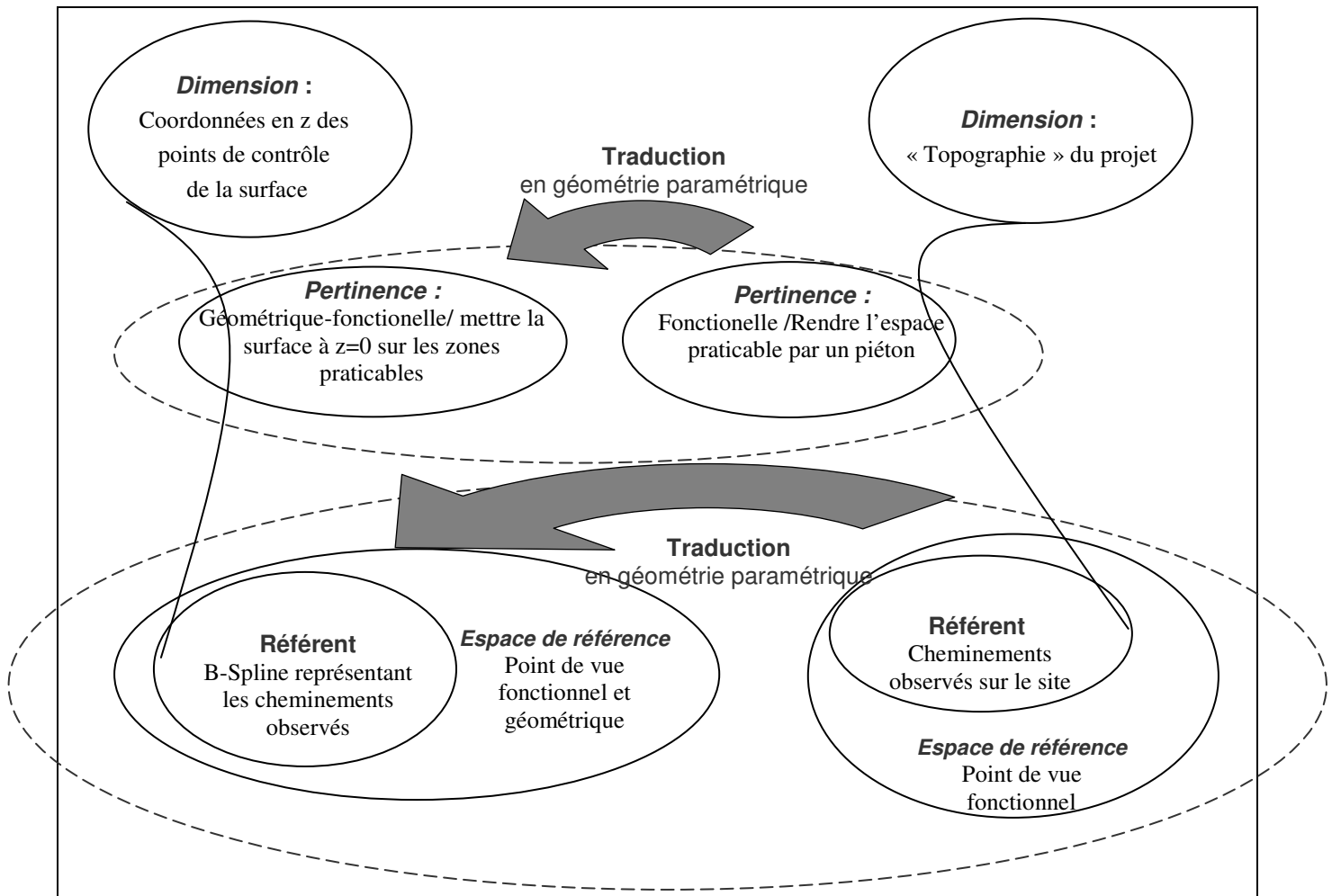
Figure 87: Opération d'appropriation de l'outil, permettant un partage d'espace de référence

L'*appropriation* de l'outil par un concepteur peut prendre la forme d'une interprétation de ces potentialités, comme dans le cas de la conception de Frank Gehry. L'interprétation qu'il a eu des potentialités de CATIA l'a mené à concevoir l'architecture par rapport à un nouvel *espace de référence* technique (cf. 6.3.2). Ce nouvel *espace de référence* technique a transformé sa conception architecturale (cf. 3.2.1.2).

#### 6.4.2.2 Opération de traduction en géométrie paramétrique

Pour décrire son activité de conception de modèles paramétriques, Hugh Whitehead, responsable du Specialist Modelling Group de Foster and Partners (cf. 3.3) pointe l'importance d'un travail qu'il nomme d'« interprétation » d'intentions architecturales en géométrie et en dépendances (Whitehead 2009b). Nous nommerons cette activité « opération de traduction en géométrie paramétrique ».

L'*opération d'interprétation pour la collaboration* est une opération visant à construire un sens personnel aux propos d'un interlocuteur (Ben Rajeb 2012, p.285). L'opération de *traduction en géométrie paramétrique* pointe l'activité de passage d'un *espace de conception* architectural à un *espace de conception* paramétrique (et vice-versa d'un *espace de conception* paramétrique à un *espace de conception* architectural). Par exemple, dans le cas du projet « Topographie », l'identification sur le site des principaux chemins piétons et la traduction de ces chemins en courbes sur Grasshopper est une forme de *traduction en géométrie paramétrique* (cf. Figure 88).



**Figure 88: Exemple d'opération d'interprétation de géométrie paramétrique dans le cas du projet « Topographie »**

Pour pointer cette activité Hugh Whitehead, comme la plupart des acteurs du SMG ou du groupe ARD, utilise le terme « interprétation » (cf. entretiens insérés en annexe pp. 327 à 340). Nous utilisons quant à nous le terme « traduction ». Au sens courant, une interprétation est l'« *Action de donner un sens personnel, parmi d'autres possibles, à un acte, à un fait, dont l'explication n'apparaît pas de manière évidente* » [cnrtl.fr/definition/]. Le terme « traduction », quant à lui, désigne le « *Fait de transposer un texte d'une langue dans une autre* » et, par analogie : la « *Transposition d'un système dans un autre* » [cnrtl.fr/definition/]. Pour nous, l'« *opération de traduction de géométrie paramétrique* » désigne à la fois l'activité de transposer une *pertinence* ou un *espace de référence* architectural en *pertinence* ou *espace de référence* paramétrique, ceci, en se basant sur une explication personnelle, la transposition littérale ne semblant pas possible.

La traduction est une part extrêmement importante du travail du Specialist Modelling Group (Whitehead 2009)<sup>81</sup> et est reconnue comme telle. Dans la pratique du SMG, l'interprétation nécessite un fort travail de collaboration avec les équipes de conception pour amener les concepteurs à expliciter leurs intentions en vue de les interpréter (Peters in Freiburger 2007).

<sup>81</sup> « *To some extent it is a support draw but in all case it is very interpretive.* » (Whitehead 2009b).

La traduction est parfois poussée jusqu'à la « *prédiction* » (Whitehead 2009)<sup>82</sup> lorsque des hypothèses de *pertinences* qui ne sont pas encore définies clairement sont faites pour être traduites et donc prévues par avance. Par exemple, pour le projet du City Hall, Whitehead décrit un des modèles paramétriques de la volumétrie (Whitehead 2009)<sup>83</sup> en insistant sur le fait que le SMG a proposé certaines transformations du modèle architectural : il a traduit les intentions de l'équipe de conception et a cherché à anticiper, ou « prédire », les futures transformations de la volumétrie des concepteurs (Whitehead 2009).

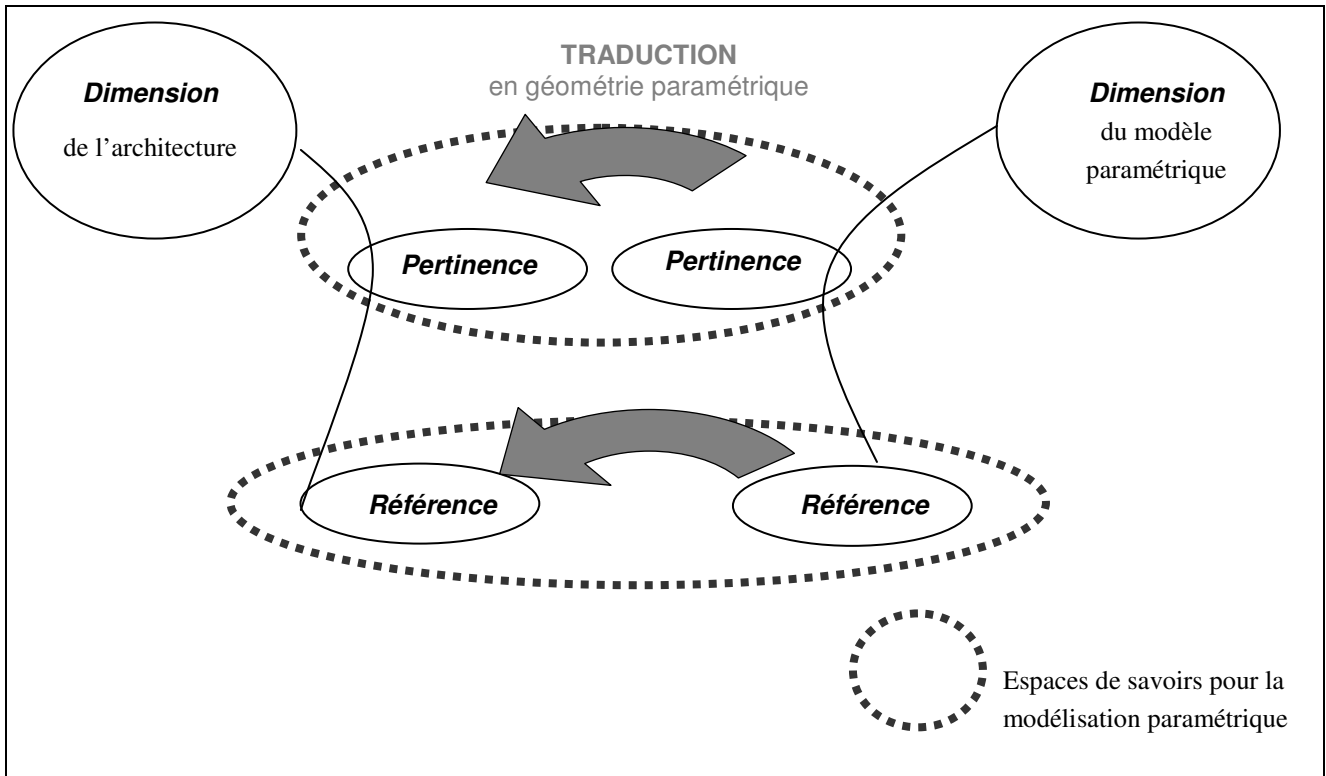
L'activité de *traduction en géométrie paramétrique* nécessite plusieurs compétences et savoirs : non seulement par rapport aux techniques relatives à la modélisation paramétrique (géométrie, programmation, maniement des modeleurs, etc.) mais surtout, selon Whitehead, par rapport à la conception architecturale (Whitehead in Kocaturk & Medjdoub 2011, p.238)<sup>84</sup>. Les compétences et les savoirs nécessaires à l'usage de la modélisation paramétrique pour la conception architecturale forment un « espace » théorique rendant possible les différentes activités mentales nécessaires à la conception de modèles paramétriques pour la conception architecturale. Dans la présente recherche, nous nommerons cet espace « espace de savoirs pour la modélisation paramétrique » (cf. Figure 89). Les savoirs, savoir-faire, savoir-être et compétences constituant cet espace seront plus particulièrement décrits dans le chapitre 8, en vue de proposer une pédagogie de la modélisation paramétrique destinée aux architectes.

---

<sup>82</sup> « *In order to make such a mechanism, we have to predict what the designers want to do, what the changes they need to make.* » (Whitehead 2009).

<sup>83</sup> « *The curvature proportions for the end are in fact total ended to the axis, so if the designer changes the length of the axis, the proportions remain. So in this way we are predicting that they may want to extend or reduce the axis, but they already set curvature proportions which they want to retain. This is an example of how we have to predict the changes they need to make.* » (Whitehead 2009).

<sup>84</sup> « *we usually like people to spend more time in the design team before they joined the SMG and I think that is always essential, because we have to be able to carry out an interpretative role and a lot of what we do is down to interpretation* » (Whitehead in Kocaturk & Medjdoub 2011, p.238).



**Figure 89: Opérations de traduction en géométrie paramétrique et espace de savoirs pour la modélisation paramétrique**

### 6.4.3 Opérations logiques

Les deux opérations mentales de logique impliquées dans la conception de modèles paramétriques en conception architecturale sont des opérations *d'induction* et de *vérification*. Ces opérations ne sont pas spécifiques à la modélisation paramétrique mais interviennent dans les mécanismes impliqués par son usage en conception architecturale.

Comme vu précédemment (*cf.* 6.2.1), on peut distinguer l'activité de conception de méta-modèles paramétriques (des environnements de modélisation ou des nouvelles fonctions), de l'activité de conception de modèles paramétriques et de l'activité de conception d'instances. Si jusqu'ici nous nous sommes centrés sur la conception de modèles paramétriques, il nous faut aborder la conception d'instances et la conception de méta-modèles paramétriques qui communiquent avec elle.

Nous l'avons vu, la conception peut porter sur des objets différents : des *unités de conception*, que sont le modèle paramétrique, le méta-modèle paramétrique et l'instance. Or l'activité de conception de modèle paramétrique semble s'attacher à ces trois *unités de conception* en aller-retour, au travers de vérifications et d'inductions successives.



### 6.4.3.1 Opération de vérification

Nous entendons par opération de « vérification », l'opération par laquelle un concepteur teste un modèle paramétrique ou un « méta-modèle » (cf. 6.2.1) par son instantiation dans des cas particuliers. Le passage à la conception d'instances est indispensable lors de la conception d'un modèle paramétrique ou d'un méta-modèle (cf. Figure 90, relations (A)). En effet concevoir une instance permet de tester et contrôler le bon fonctionnement et la cohérence de ces derniers. Par exemple, quand la powercopy « calepinage » du projet Radical Reproduction est conçue, elle est ensuite transformée pour être instanciée et testée dans un cas particulier de surface. Ce passage de la conception du modèle paramétrique à la conception de l'instance permet de vérifier la viabilité du système conçu. De la même façon, la conception d'un plugin nécessite des tests de validation dans un projet *via* la conception de modèles paramétriques spécifiques (cf. Figure 90, relations (B)). Par exemple, l'usage par l'agence DECODE du plugin GeoGym sur un modèle paramétrique particulier, a mis au jour un certain nombre de bugs et de lacunes de l'outil. Une collaboration étroite entre DECODE et le développeur de GeoGym a permis aux collaborateurs des deux bords à la fois d'implémenter et de compléter le plugin, mais aussi de l'utiliser au mieux et d'affiner le modèle paramétrique développé.

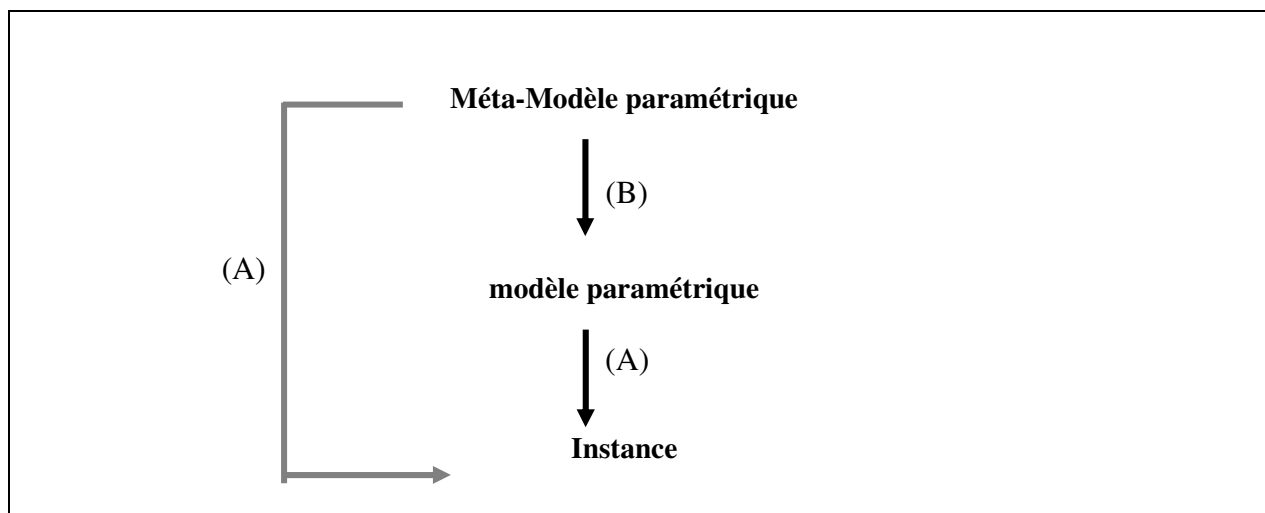
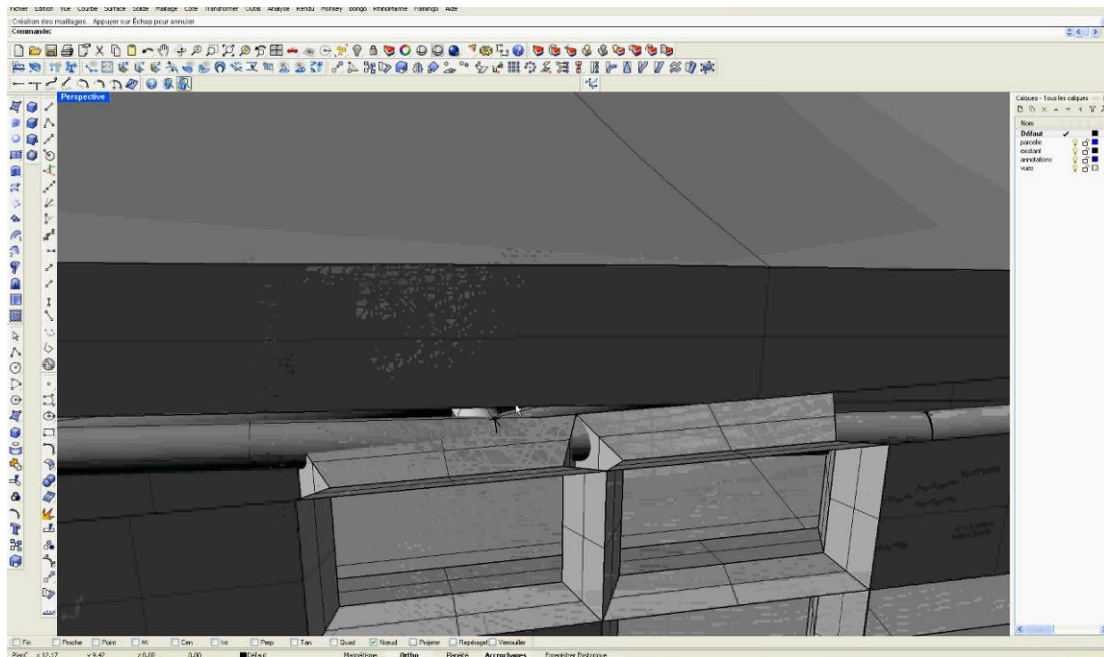


Figure 90 : Opérations de vérification opérant par le passage d'une *unité de conception* à l'autre

Cette opération de « test » du modèle paramétrique vise à évaluer la *viabilité* du mécanisme informatique mis en place, du domaine de solutions défini par le modèle numérique génératif ainsi qu'à visualiser le panel de formes défini par celui-ci. Par exemple, dans le cas du projet « Topographies » (cf. Figure 68), des tests fréquents des instances de la surface générées et de leurs impacts dans le site ont permis de mettre en évidence des défauts du modèle en cours de conception. En particulier, si le dimensionnement de la surface par rapport aux cheminements a été précoce, son dimensionnement en fonction du voisinage et de son impact dans le site a été provoqué et effectué grâce à ces allers et retours entre conception du modèle paramétrique et conception de l'instance « topographie ».

De même, par exemple, dans le cas du processus de projet du binôme H3 de l'Expe2, une phase d'instanciation de leur modèle a montré un problème de raccord entre une toiture et une paroi (cf. Figure 91).

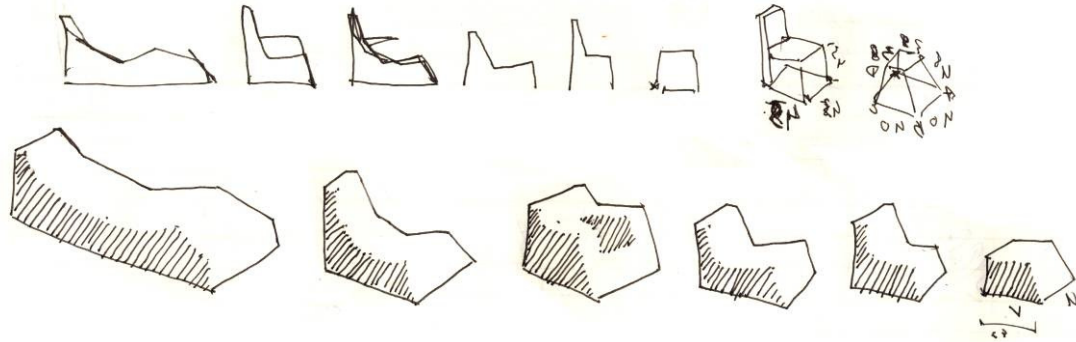


**Figure 91 : Contrôle de la géométrie créée au travers d'une instance sur Rhinocéros : capture d'écran issue du film de l'expérimentation du binôme H3 (Expe2)**

L'exploration de variables donne lieu à des séries d'instanciations du modèle. Ces instanciations sont évaluées en fonction de la (ou des) *pertinence(s)* mise(s) en œuvre dans les opérations de conception du modèle paramétrique. Ces instanciations peuvent rester temporaires ou être figées et exportées pour d'autres utilisations de la géométrie (évaluations, productions de représentations de rendu, etc.).

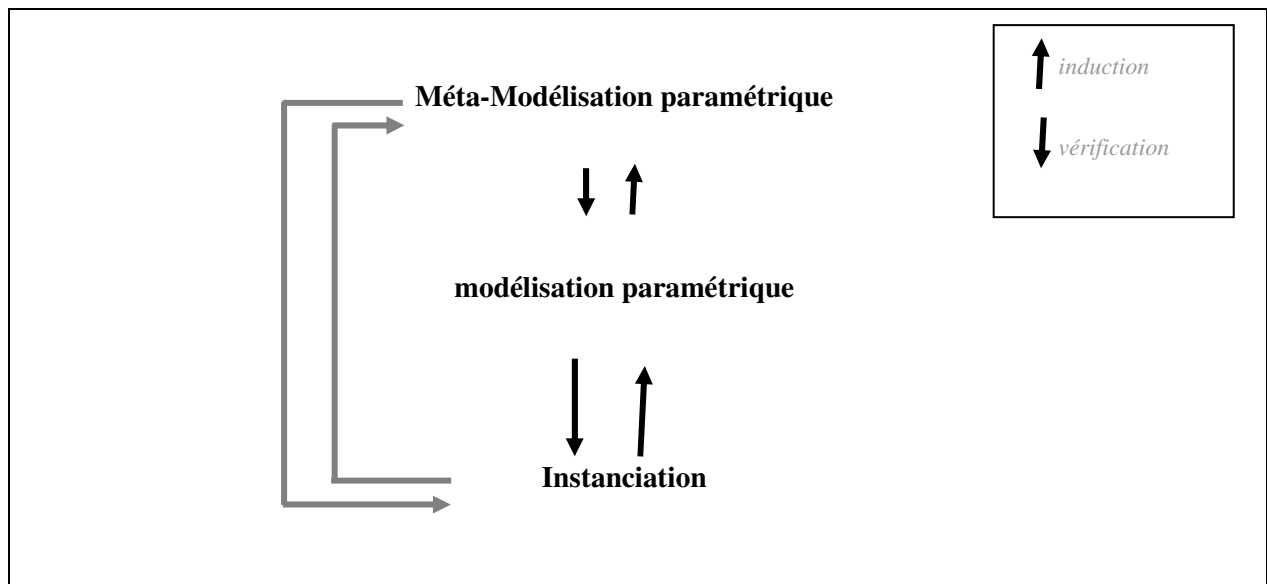
### 6.4.3.2 Opération d'induction

Le passage d'une *unité de conception* à l'autre n'est pas toujours « descendant » et ne vise pas seulement la vérification (cf. Figure 90). Les relations peuvent aussi aller vers la généralisation et l'abstraction d'un cas particulier en une unité de conception différente. On observe un tel mécanisme dans le cas du projet « Observer » développé par Martial Marquet par son Master Spécialisé de l'ENSCI. Dans le cadre de la conception de mobiliers pour les musées, Marquet a initié son processus par l'exploration de postures spécifiques à certaines fonctions (observer de loin, observer de près, s'appuyer, s'allonger, etc.). Ces postures ont été cristallisées dans des objets, comme « chaise », « chaise longue », « chauffeuse » etc. visibles dans la Figure 92. Ces objets correspondent à une unité de conception d'instances, à partir de laquelle un système a été conçu pour générer une famille de solutions comprenant ces cas particuliers.



**Figure 92: Esquisses exploratoires « Observer», source: Martial Marquet**

Dans ce dernier cas, le passage d'une *unité de conception* à l'autre implique l'induction d'un système général à partir d'une famille de cas particuliers (cf. Figure 93, flèches ascendantes).

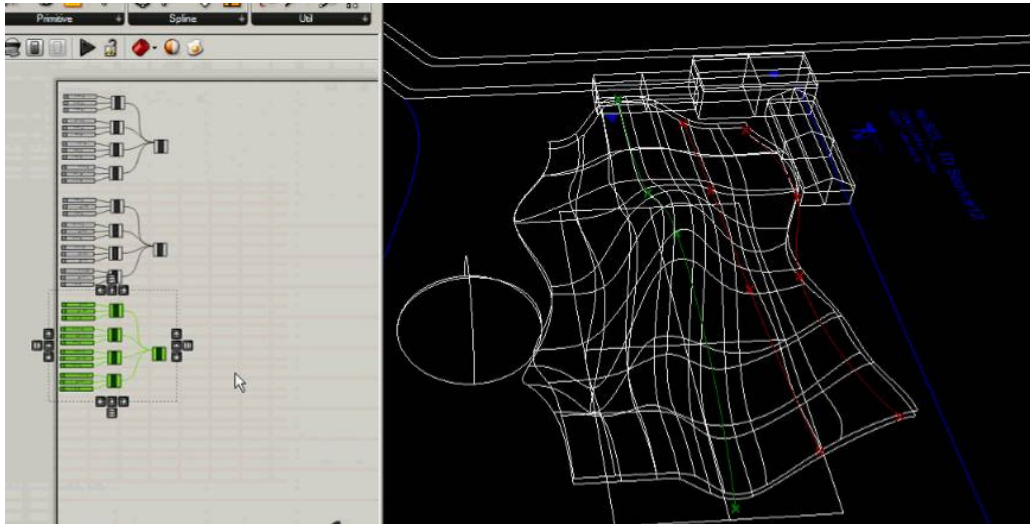


**Figure 93 : Opérations de vérification et d'induction opérant par le passage d'une unité de conception à l'autre**

Ces articulations d'*unités de conception* impliquées dans la conception de modèle paramétrique, pointent l'usage d'opérations logiques spécifiques de « vérification » et d'« induction ».

Nous nommons «opération d'induction» l'opération mentale que nous reprenons au champ de la logique et qui consiste à inférer un système général à partir de cas particuliers.

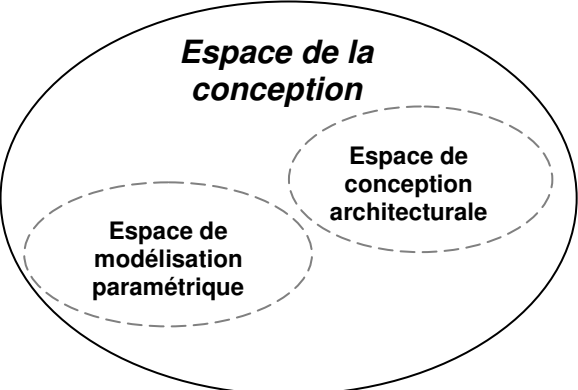
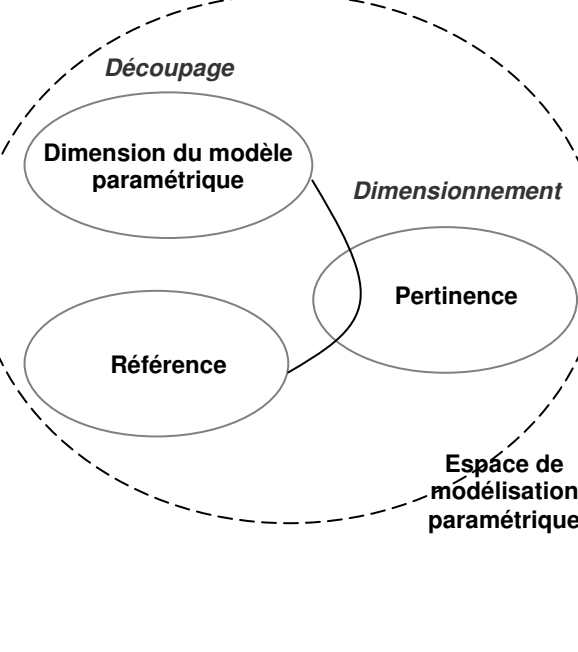
L'opération d'induction est fréquente lors de la conception de modèles paramétriques. Par exemple, lors du processus de conception d'un projet de piscine dans l'Expe1, le groupe C1 (cf. annexe p.389) est passé par une phase d'exploration de formes sur Rhinocéros afin de concevoir et de s'accorder sur des intentions. Cette forme a ensuite été approchée par une modélisation sur Grasshopper. Les étudiants ont décidé de généraliser la production de la forme préalablement conçue, en construisant une surface s'appuyant sur des courbes de sections. La forme initiale (en blanc à droite sur la Figure 94) a été discrétisée en points définis par des coordonnées absolues (à gauche sur la Figure 94).



**Figure 94 : Premières étapes de modélisation du groupe C1 (capture d'écran issue du film de l'expérimentation)**

## 6.5 Synthèse des caractérisations cognitives de la conception de modèles paramétriques en conception architecturale

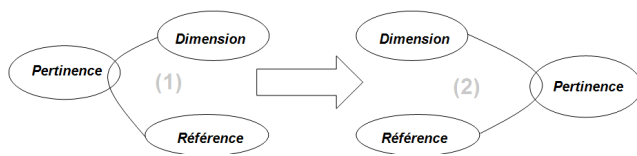
Une synthèse des différentes caractérisations cognitives de l'usage de la modélisation paramétrique en conception architecturale est effectuée dans le Tableau 31.

| Opérations élémentaires de conception de modèles paramétriques  |   |
|---|---|
|  <p>Le diagramme illustre l'<b>Espace de la conception</b> (cercle plein) qui englobe deux sous-espaces : l'<b>Espace de modélisation paramétrique</b> (cercle pointillés) et l'<b>Espace de conception architecturale</b> (cercle tirets).</p>   | <p>Opérations de conception architecturale et opérations de conception de modèles paramétriques peuvent être interrogées comme des opérations de la conception, c'est-à-dire comme opérations cognitives d'attribution de mesures.</p>  |
|  <p>Le diagramme illustre l'<b>Espace de modélisation paramétrique</b> (cercle tirets) qui englobe quatre éléments : <b>Découpage</b>, <b>Dimension du modèle paramétrique</b>, <b>Pertinence</b> et <b>Référence</b>. Des lignes relient 'Dimension du modèle paramétrique' à 'Dimensionnement' et 'Pertinence', et 'Référence' à 'Pertinence'. Le terme 'Dimensionnement' est également présent à l'intérieur de l'espace.</p> | <p>La modélisation paramétrique implique principalement :</p> <ul style="list-style-type: none"> <li>- des opérations de <i>découpage</i> : c'est-à-dire d'organisation du modèle pour définir ce sur quoi va porter la mesure ;</li> <li>- des opérations de <i>dimensionnement</i> : c'est-à-dire de mise en relation d'un <i>espace de référence</i> et d'une <i>dimension</i> via une <i>pertinence</i> ;</li> <li>- des opérations de <i>référenciation</i> : c'est-à-dire de construction d'un <i>espace de référence</i> dans le cadre d'une opération de <i>dimensionnement</i>.</li> </ul> |
| Mise en œuvre des opérations élémentaires de conception de modèles paramétriques  |   |
| <p>Une <i>unité de conception</i> est un ensemble de <i>dimensions</i>, c'est-à-dire d'objets auxquels on attribue des mesures. Trois <i>unités de conception</i> récurrentes dans la conception de modèle paramétrique peuvent être</p>  |   |

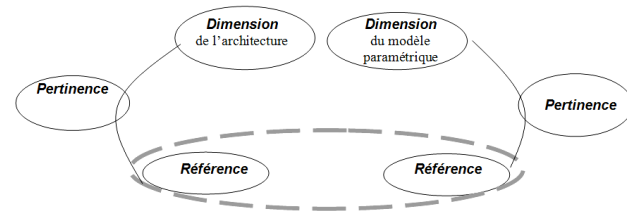
identifiées : celle de l'instance, celle du modèle paramétrique et celle du méta modèle paramétrique. *L'unité de conception du modèle paramétrique* est celle qui est interrogée dans cette thèse, même si les deux autres lui sont très liées (cf. opérations de logiques impliquées).

La *pertinence* technique est souvent *dominante* et *structurante* dans les cas de conception des modèles paramétriques analysés. Néanmoins, cette *pertinence* n'est ni exclusive (elle peut être associée à d'autres pertinences dominantes, comme la pertinence optique dans le cas de la pratique de Gehry Partners) ni incontournable (des opérations de modélisation paramétrique liées à la conception architecturale peuvent opérer sans qu'elles ne soient sollicitées).

### Relation entre opérations

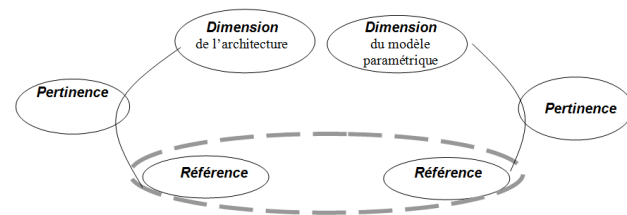


Des relations temporelles en *cascade* ou en *relais* peuvent être observées entre opérations de conception de modèles paramétriques et opérations de conception architecturale.



Des relations de partage *d'espace de références* ou de partage de *pertinences* sont observées entre opérations de conception de modèles paramétriques et opérations de conception architecturale.

### Opérations tierces mises en œuvre

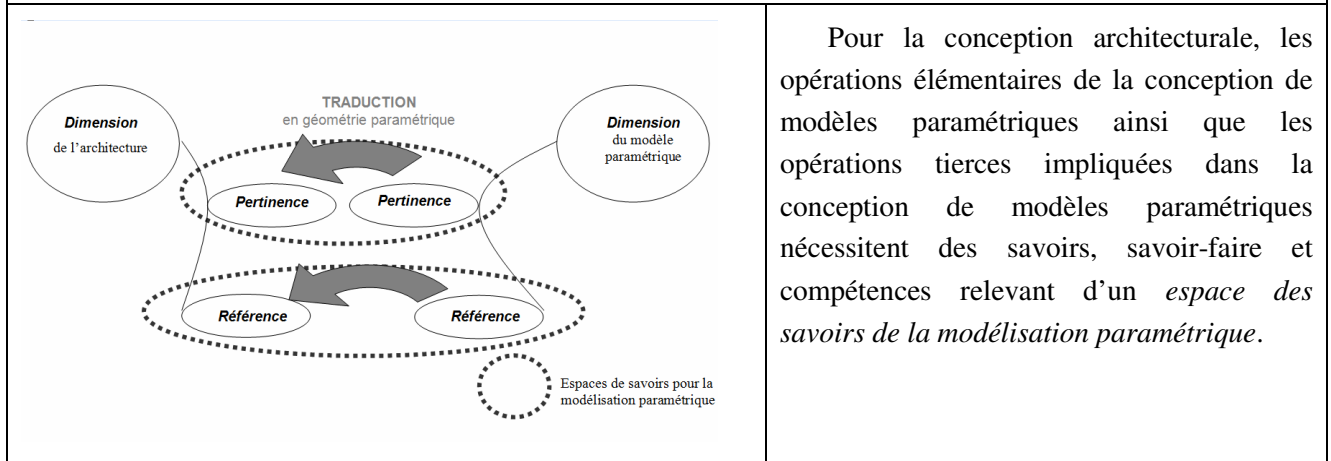


Construction d'un espace de référence par collaboration ou par usage de l'outil

Des *Opérations pragmatiques* de conception de la situation de conception peuvent être observées : l'*opération de collaboration* « *d'interprétation* » et l'*opération d'usage de l'outil* « *d'appropriation* ». Ces deux opérations permettent, entre autres, à un ou plusieurs concepteurs de se construire des *espaces de référence* propres à la conception de modèles paramétriques. Ces *espaces de référence* peuvent être partagés avec les *espaces de référence* d'opérations de conception architecturale.

|  |   |
|--|---|
|  | <p>L'opération de <i>traduction en géométrie paramétrique</i> désigne l'activité d'interprétation et de traduction de <i>pertinences</i> et d'<i>espaces de référence</i> de conception architecturale en <i>pertinences</i> et <i>espaces de référence</i> de conception de modèles paramétriques.</p>   |
|  | <p>Des opérations logiques d'induction et de vérification peuvent être observées. Ces opérations correspondent à un passage d'une <i>unité de conception</i> à l'autre, visant à : - contrôler la viabilité ou la cohérence (opération de <i>vérification</i>) et, -induire un système général à partir de cas particuliers (opération d'<i>induction</i>).</p> |

## Vers la définition d'un espace des savoirs de la modélisation paramétrique en conception architecturale



Pour la conception architecturale, les opérations élémentaires de la conception de modèles paramétriques ainsi que les opérations tierces impliquées dans la conception de modèles paramétriques nécessitent des savoirs, savoir-faire et compétences relevant d'un *espace des savoirs de la modélisation paramétrique*.

**Tableau 31: Synthèse des caractérisations cognitives de l'usage de la modélisation paramétrique pour la conception architecturale**

Ces connaissances sont autant d'outils théoriques permettant de décrire la conception de modèles paramétriques en conception architecturale. Ils constituent le principal apport de notre recherche et nous permettent d'interroger également :

- La pertinence et les spécificités de la réutilisation de modèles paramétriques déjà conçus comme aide à la conception de modèles paramétriques en conception architecturale (Chapitre 7);
- Le développement d'une pédagogie de la modélisation paramétrique destinée aux futurs architectes (Chapitre 8).



## 7 Analyse architecturologique de l'usage de la méthode des Patterns

Grâce à l'appareillage théorique de l'architecturologie, nous avons dans le chapitre précédent caractérisé les opérations cognitives impliquées dans la conception de modèles paramétriques en conception architecturale. Dans ce chapitre, nous analysons l'usage de certaines des assistances proposées aux architectes pour les aider à construire des modèles paramétriques, aides que nous avons présentées dans le Chapitre 4. Nous abordons particulièrement l'usage des par patterns proposés par Robert Woodbury (cf. 4.1.2) au travers des samples. L'usage de patterns constitue-t-il une assistance pertinente à la conception architecturale ? est la question abordée par ce Chapitre.

### Plan du chapitre :

|            |  |            |
|------------|--|------------|
| <b>7.1</b> | <b>Expérimentations menées .....</b>   | <b>236</b> |
| 7.1.1      | Construction des Samples pour les expérimentations   | 236        |
| 7.1.2      | Samples utilisés lors des expérimentations   | 240        |
| 7.1.3      | Modalités d'appropriation des samples  | 242        |
| <b>7.2</b> | <b>Spécificités cognitives de l'usage de Samples pour la conception de modèles paramétriques en conception architecturale .....</b>        | <b>244</b> |
| 7.2.1      | Exploration de bibliothèques et choix d'un sample : prise de décision et attribution de mesure à un modèle paramétrique                    | 244        |
| 7.2.2      | Stratégies d'appropriation de samples : <i>dimensionner</i> un modèle paramétrique conçu par un autre                                      | 245        |
| <b>7.3</b> | <b>Discussion de l'usage des patterns et des samples comme aide à la modélisation paramétrique pour la conception architecturale .....</b> | <b>257</b> |

## 7.1 Expérimentations menées

Nous avons vu précédemment l'aide à la modélisation paramétrique proposée par Robert Woodbury : la méthode des patterns (cf. 4.1.2). Cette méthode est potentiellement puissante (cf. 4.1) et médiatisée par patterns appliqués : les samples (cf. 4.2). Un sample est un fragment de modèle paramétrique permettant de résoudre un problème technique récurrent dans la conception de modèles paramétriques en conception architecturale. Nous avons vu dans le chapitre 4 que de nombreux samples sont diffusés par une communauté d'entraide active (cf. 4.2.3).

Woodbury relève le besoin de produire des recherches interrogeant l'aide des patterns à la modélisation paramétrique en conception architecturale (Woodbury 2010, p.188)<sup>85</sup>. C'est cette invitation que suit cette recherche. Des expérimentations ont été mises en place pour constituer un corpus d'analyse de cas d'usages de patterns appliqués en conception architecturale. Cette méthode de recueil de données est décrite précisément dans la section 5.4 précédente (cf. p. 170). Les données recueillies ont été analysées grâce à l'appareillage théorique de la recherche (cf. 5.1).

Le dispositif des expérimentations semble avoir fonctionné pour la constitution de cas d'usages de patterns appliqués : les samples. Les étudiants ont majoritairement « joué le jeu » et la plupart des données ont été recueillies. Des samples ont été proposés aux étudiants lors des expérimentations et leur ont permis de produire rapidement des formes. Même les étudiants ayant déjà une expérience de la modélisation paramétrique ont utilisé des samples. Peu de binômes n'a pas du tout utilisé de Sample. Ceux qui n'en ont utilisé aucun ont exprimé le fait de n'avoir trouvé aucun Sample adapté à leurs intentions.

Si le corpus de données recueillies permet d'analyser l'usage de patterns appliqués (les samples) pour la conception de modèle paramétrique en conception architecturale, il n'a pas été possible d'analyser l'usage de patterns non appliqués, tels que proposés par Robert Woodbury, ceci même après plusieurs adaptations des cours et des protocoles associés aux expérimentations. Lors des expérimentations, c'est l'application des patterns dans des cas particuliers (les samples) qui a rendu effective l'aide des patterns. L'effectivité de l'aide des patterns sera questionnée plus précisément dans la suite du texte.

### 7.1.1 Construction des Samples pour les expérimentations

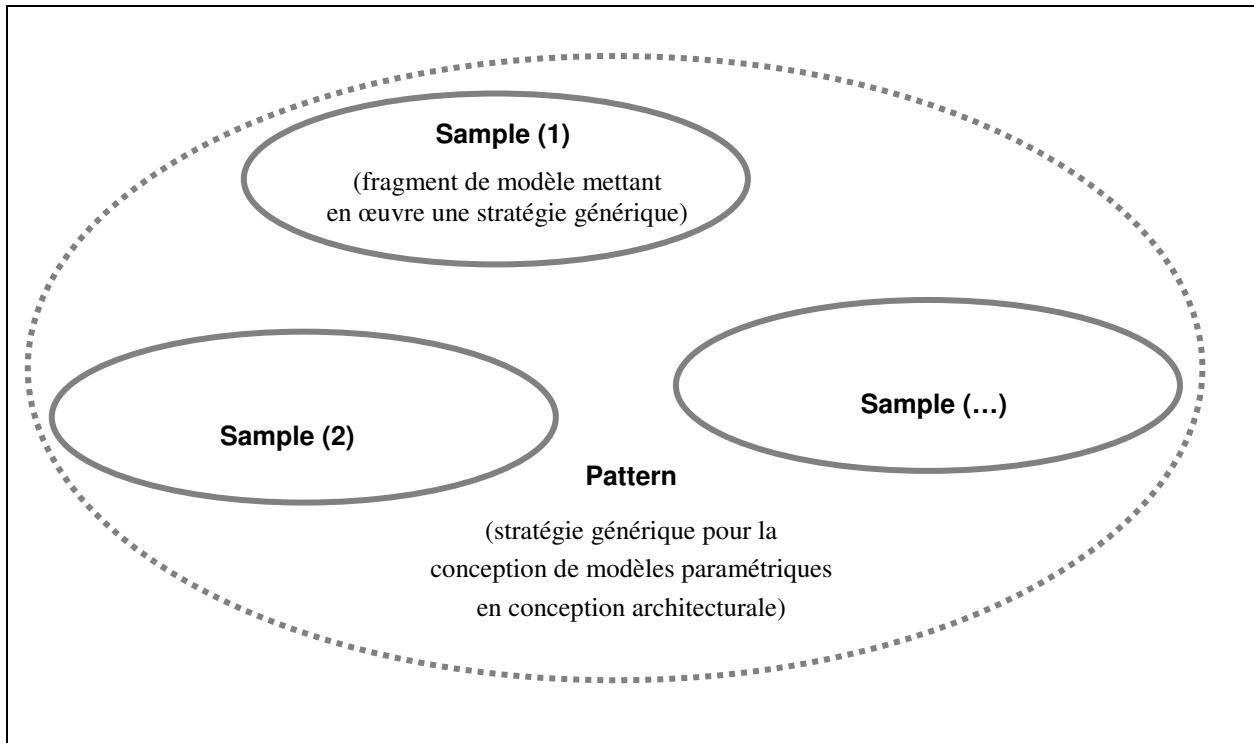
Les expérimentations ont été menées sur le modeleur paramétrique Grasshopper. Après une analyse de ce modeleur (cf. 2.3.1) et des aides à la modélisation paramétrique existantes (cf. 4.2), ce logiciel semble le plus facile d'accès pour des étudiants en architecture ou des architectes n'ayant aucun pré-requis en programmation informatique.

Dans le cadre des expérimentations, des samples ont été construits pour assister l'usage de Grasshopper pour la conception de modèles paramétriques en conception architecturale. Comme nous

---

<sup>85</sup> « *In the place of firm conclusions, I hypothesize that patterns can help design work and present several arguments supporting this hypothesis.* » « *The truth is that these [patterns] are propositions to be tested by future research.* » (Woodbury 2010, p.188).

l'avons défini précédemment (cf. 4.1.2), des samples sont des fragments de modèles paramétriques, mettant en œuvre des stratégies de conception spécifiques : les patterns. Les patterns sont des stratégies génériques de conception de modèles paramétriques. Les samples sont l'application de ces stratégies à des cas de conception de modèles paramétriques spécifiques. Un sample est un pattern appliqué à des cas particuliers. A un même pattern peuvent correspondre une multitude de samples (cf. Figure 95).



**Figure 95 : Relation entre patterns et samples**

Les Samples construits lors des expérimentations répondent à deux critères :

- ce sont des applications de patterns de Woodbury ;
- ils répondent à des problèmes techniques propres à la modélisation paramétrique et récurrents en conception architecturale (comme la gestion d'arbres de données (cf. 2.1.3), l'évaluation de l'ensoleillement d'une surface, l'utilisation d'une fonction d'optimisation, etc.).

Les samples construits pour les expérimentations sont décrits succinctement dans le Tableau ci-dessous, selon les deux critères cités (patterns appliqués et solutions techniques traitées). Ces samples sont organisés ici en familles de solutions techniques.

| <b>Famille et noms des samples</b>   | <b>Patterns appliqués</b> | <b>Questions techniques abordées</b> |
|--------------------------------------|---------------------------|--------------------------------------|
| <i>Gérer des arbres de données :</i> |                           |                                      |
| <b>extrait_1item_liste4</b>          | Place Holder              | Arbres de données                    |
| <b>Extrait_valeur_min-max</b>        | selector                  | idem                                 |
| <b>Mapping_list</b>                  | mapping                   | idem                                 |
| <i>Créer des points</i>              |                           |                                      |

|   |                  |  |
|---|------------------|--|
| <b>Creer_trame-pts_UV</b>                     | Point Collection | Créer un module sur une surface                                    |
| <b>Creer_pt_UV</b>                            | Controller       | Comprendre la notion de « domaine » d'une surface                  |
| <b>creer_liste-pts_sous-srf_UV</b>            | Point Collection | Créer un module sur une surface                                    |
| <b>Creer_pts-aleatoires_sur-trame-UV</b>      | Point Collection | Générer des objets aléatoires                                      |
|   |                  |  |
| <i>Créer des courbes</i>                      |                  |  |
| <b>Creer_crv_math-spirale</b>                 | Point Collection | Construire une courbe à partir de sa fonction paramétrique         |
| <b>creer_crv_math-spirale3D</b>               | Point Collection | Idem   |
| <b>Creer_crv_math-balle-tennis</b>            | Point Collection | idem   |
| <b>creer_courbes-sections_surfaces-sweep2</b> | Reporter         | Positionner des plans par rapport à une surface                    |
| <b>creer_crv_voronoi</b>                      | Point Collection | Créer un diagramme de Voronoi                                      |
|   |                  |  |
| <i>Créer des surfaces</i>                     |                  |  |
| <b>Creer_Surface-Projetee_RandomDelaunay</b>  | Point Collection | Créer un diagramme de Delaunay                                     |
| <b>creer_surface-projetee_attracteur</b>      | Controller       | Faire réagir une trame de points à leurs distances avec des objets |
| <b>creer_srf_niveaux</b>                      | Reporter         | Évaluer la surface constructible d'un volume                       |
| <b>Créer_surfaces_math-shell</b>              | Point Collection | Créer une surface à partir de son équation paramétrique            |
|   |                  |  |
| <i>Créer motif</i>                            |                  |  |
| <b>Creer_motif-maillage</b>                   | Place Holder     | Créer un module sur une surface                                    |
| <b>Créer_Motif-reactif_attracteurUV</b>       | Controller       | Faire réagir des objets situés sur une surface à un point          |
| <b>Creer_motif-morphing</b>                   | Mapping          | Répéter un objet régulièrement sur une surface                     |
|   |                  |  |
| <i>Evaluer</i>                                |                  |  |
| <b>Evaluer_planeite-surface</b>               | Reporter         | Évaluer la planité d'une   |

|                           |          |  |
|---------------------------|----------|--|
|                           |          | surface  |
| <b>Evaluer_écart-type</b> | Reporter | Évaluer l'écart type d'une liste de valeurs                            |
| <b>Héliodon</b>           | Reporter | Créer un héliodon en fonction des spécificités géographiques d'un site |
|                           |          |  |

**Tableau 32 : Samples construits pour les expérimentations**

Les samples proposés ici semblent particulièrement récurrent dans l'usage de la modélisation paramétrique en conception architecturale (cf. Chapitre 3) tout en restant encore suffisamment généraux pour être appropriés dans des situations différentes de conception architecturale.

Par exemple, le sample *Extrait\_valeurs\_min\_max* est une application du pattern *selector*. Le pattern *selector* : « *extrait les éléments d'une collection ayant des propriétés particulière* »<sup>86</sup> [designpatterns]. Le sample *Extrait\_valeurs\_min\_max* permet d'extraire des objets associés aux valeurs les plus élevés ou les plus basses leur étant associés. Il est une application du pattern *selector*. Il permet de résoudre un problème récurrent de la modélisation paramétrique qui est d'extraire un objet spécifique à partir d'une ou plusieurs listes d'objets, en manipulant la structure des données d'un modèle. Ce sample est un fragment de modèle opérant une action, mais il peut être utilisé à diverses fins. Par exemple il peut être utilisé pour extraire les points les plus proches d'un objet de référence (et ainsi participer à la mise en œuvre d'une *échelle de voisinage* par exemple), ou encore pour extraire une surface dont l'aire est la plus importante (et ainsi participer à la mise en œuvre d'*échelle économique* ou *technique* par exemple).

Un autre exemple est le sample *Surface\_projete attracteur*. Ce sample est l'application du pattern *controller*. Le pattern *controller* : « *dirige une partie de modèle au travers d'un modèle séparé* »<sup>87</sup> [designpatterns]. Le sample *Surface\_projete attracteur* permet de modéliser une surface réagissant à la proximité d'objets comme des points ou des courbes. Plus la surface est proche d'un point, plus elle se déforme. C'est sur ce sample que ce sont appuyés les étudiants ayant développé le projet « Topographie », étudié précédemment (cf. 6.1.2). Il a donc permis la mise en œuvre divers *échelles*, telles qu'*optique*, de *voisinage*, *fonctionnelle*, etc.

Les samples développés pour les expérimentations ont évolués :

- dans leurs contenus : certains samples ont été transformés, parfois à la demande des étudiants ;
- dans leurs formes, c'est-à-dire dans la présentation de la collection de fichiers aux étudiants.

Ces évolutions et leurs raisons sont développées tout au long du texte et, plus particulièrement, dans le Chapitre 8 qui s'intéresse à la construction d'une pédagogie de la modélisation paramétrique. La collection de samples et leurs évolutions sont décrites et détaillées dans l'annexe 3 (cf. p. 417).

<sup>86</sup> "Select members of a collection that have specified properties." [designpatterns]

<sup>87</sup> "Control (a part of) a model through a simple separate model." [designpatterns]

En plus de ces samples construits pour les expérimentations, des références de catalogues de samples en ligne (cf. 4.2.2) ont été fournies aux étudiants. Les étudiants ont donc également exploré et utilisé des samples développés par des acteurs extérieurs à l'expérimentation.

### 7.1.2 Samples utilisés lors des expérimentations

Finalement, seul un nombre réduit de Samples a été sollicité par rapport à ceux mis à disposition des étudiants. Le Tableau 33 rend compte de la fréquence du recours aux samples proposés.

|  |             |
|--|-------------|
| <i>Gérer des arbres de données :</i>   |             |
| extrait_1item_liste4                   | 0           |
| <i>Créer des points</i>                |             |
| Creer_reseau-pts_UV                    | 0           |
| Creer_pt_UV                            | 0           |
| creer_liste-pts_sous-srf_UV            | 0           |
| Creer_pts-aleatoires_sur-trame-UV      | 0           |
| Creer_pts-aleatoires                   | 0           |
| <i>total</i>                           | <i>0</i>    |
| <i>Créer des courbes</i>               |             |
| Creer_crv_math-spirale                 | 0           |
| creer_crv_math-spirale3D               | 2           |
| Creer_crv_math-balle-tennis            | 0           |
| creer_courbes-sections_surfaces-sweep2 | 0           |
| creer_crv_voronoi                      | 2           |
| <i>total</i>                           | <i>4</i>    |
| <i>Creer des surfaces</i>              |             |
| Creer_Surface-Projetee_RandomDelaunay  | 0           |
| creer_surface-projetee_attracteur      | 3           |
| creer_srf_niveaux                      | 0           |
| Créer_surfaces_math-shell              | 0           |
| OCP Wave                               | 1           |
| <i>total</i>                           | <i>3+1</i>  |
| <i>Créer motif</i>                     |             |
| Creer_motif-maillage                   | 8           |
| Créer_Motif-reactif_attracteurUV       | 2           |
| Creer_motif-morphing                   | 7           |
| Hexagone                               | 2           |
| Dispatch Panels                        | 4           |
| <i>total</i>                           | <i>17+6</i> |
| <i>Evaluer</i>                         |             |
| Evaluer_planeite-surface               | 0           |

#### Légende :

En gris : patterns venant de site autres que celui du cours.

**Notes :** Les catégories ont été établies lors du cours. Les patterns extérieurs ont été classés a posteriori en fonction de ce qu'ils effectuent.

|                    |   |
|--------------------|---|
| Evaluer_ecart-type | 0 |
| Heliodon           | 2 |
| <i>total</i>       | 1 |

**Tableau 33: Fréquence des samples utilisés lors des expérimentations 1 à 3**

Le Tableau 33 montre que les samples de « motifs » sont les plus utilisés. Ces samples correspondent au Pattern place holder (*cf.* 4.1.2). Cette surreprésentation du Pattern place holder vient probablement de la facilité d'usage de ces samples. Ce Pattern permet de propager un motif sur une surface existante et les samples correspondants peuvent être utilisés sans une grande maîtrise de Grasshopper et pour des intentions architecturales variées.

Les samples des catégories « gestion d'arbres de données » et « création de points » n'ont pas été utilisés du tout. La plupart du temps, créer des points ou gérer des données était improvisé à partir des cours ou des fonctionnalités connues sur Grasshopper. Ces samples sont relativement difficiles à utiliser car ils ne permettent pas d'obtenir directement une géométrie (contrairement aux samples de surfaces ou de motifs par exemple). Ils doivent être associés à d'autres éléments.

| <i>combinaisons</i> |             |                   |                    | <i>total</i> |
|---------------------|-------------|-------------------|--------------------|--------------|
| Evaluer             | Créer motif | Créer des courbes | Créer des surfaces |              |
| o                   | X           | o                 | X                  | 1            |
| o                   | o           | o                 | o                  | 1            |
| o                   | plusieurs   | o                 | o                  | 4            |
| o                   | o           | X                 | o                  | 3            |
| o                   | plusieurs   | o                 | x                  | 2            |
| o                   | plusieurs   | x                 | x                  | 1            |
| x                   | x           | o                 | o                  | 1            |
| o                   | x           | o                 | o                  | 2            |

Légende :

X = un sample

O = aucun sample de ce type

plusieurs = plusieurs samples de ce type

**Tableau 34: Combinaisons de samples mises en œuvre durant les expérimentations**

Plusieurs combinaisons de samples ont été créées lors des expérimentations (*cf.* Tableau 34). Par exemple, les samples de motifs sont parfois combinés entre eux (*cf.* Tableau 34), jamais pour créer un nouveau motif, mais plutôt pour traiter différentes zones du projet. Cet usage semble un peu « scolaire » et destiné à répondre à l'exercice du protocole (*cf.* 5.4) qui imposait d'utiliser au minimum trois samples. Malgré le succès des samples de motifs, les samples de surfaces ne sont pas très utilisés. Les samples de motifs ont, en fait, été utilisés sur des surfaces construites à partir de zéro sur Grasshopper et non en combinaison avec des samples de surfaces. Finalement les combinaisons de samples que l'on observe dans les cas de l'expérimentation sont assez peu diversifiées (*cf.* Tableau 34).

### 7.1.3 Modalités d'appropriation des samples

Les modalités d'appropriation des samples sont très variées (cf. Tableau 35). Le Tableau 35 rend compte de quelques cas significatifs d'appropriations de samples observés lors des Expe 1 et 2. Ces modalités sont l'objet d'une analyse architecturologique particulière car elles paraissent être un moyen par lequel l'étudiant donne de la *mesure* à un modèle préconçu, c'est-à-dire par lequel il conçoit son propre modèle paramétrique à partir de fragments préconçus. Cette analyse est développée dans le paragraphe suivant (7.2).

| <i>grpe</i> | <i>Samples impliqués</i>   | <i>Type d'appropriation</i>  |
|-------------|--|--|
| C1          | - Panel-Dispatch   | Copier-coller pur et simple. Aucune modification du fichier d'origine. Probable travail en amont pour la préparation de la surface d'entrée.   |
| D1          | - Srf-Projetée-Delaunay<br>- crv-Voronoi                                     | Copier-coller. Exploration des samples jusqu'à complète déformation de ceux-ci. Des samples n'ont été gardés que des parties « anecdotiques » (très réduites et ne constituant pas la part la plus représentative du Pattern). |
| A3          | - Heliodon   | Copier-coller.   |
|             | - Panel-dispatch   | Copier-coller avec exploration et modification du samples. Motif détourné pour, de paroi, modéliser des « cabines de bains ».  |
|             | - VB script du cours   | Copier coller d'un script développé en cours et approprié lors d'une séance précédente.  |
| E3          | - Panel-dispatch   | Copier coller. Avec manipulation des variables pour obtenir des surfaces satisfaisantes en référence au moucharabîé.   |
| G3          | - Panel-dispatch   | Copier coller. Avec choix comme données d'entrée de surfaces modélisées sur Grasshopper. Les variables n'ont pas été modifiées, ce qui donne lieu à des maladdresses.  |
| A2          |  | Reproduction manuelle du Sample grâce au site du cours.  |
| B2          | -Créer-srf-projetée-attracteur<br>- créer-morphing<br>- Créer-motif-maillage | Copier coller.   |
| B3          | Créer motif attracteur   | Copier coller. Appropriation des sorties du Pattern par un retraitement plus complexe du motif (extrusion puis booléen).   |

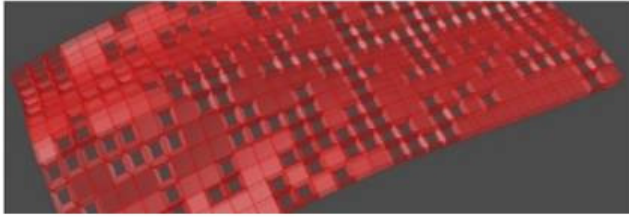
**Tableau 35: Extrait de l'analyse des modalités de reprise de samples lors des Expe1 et 2**

Lors des Expe 1 et Expe 2, un sample, appelé Panel\_Dispatch, a été choisi de façon récurrente par les étudiants. L'utilisation d'un même sample pour concevoir des modèles paramétriques spécifiques à différents projets architecturaux nous a beaucoup intéressée. Les différentes appropriations de ce sample ont particulièrement été analysées et le protocole de l'Expe 3 a été adapté pour demander aux étudiants d'utiliser obligatoirement ce sample. Cela a permis de multiplier les cas d'appropriation d'un même sample et d'évaluer la capacité des différents binômes à comprendre et à s'approprier un modèle qu'ils n'avaient jamais vu.

Ce sample appelé Panels-dispatch (cf. Figure 96) a été conçu par l'équipe de Co-de-It et est mis au téléchargement sur leur site [co-de-it]. Ce sample génère des panneaux aléatoires à partir d'une surface de référence et de variables permettant de contrôler la hauteur des panneaux ainsi que leurs densités.



### Panels dispatch



This definition allows you to tessellate a surface with two different panel types dispatched through random sorting.

[Panels dispatch\\_Co-de-iT\\_GH06019.zip](#) < Grasshopper definition (for GH v06.0019)

[Panels dispatch\\_Co-de-iT\\_3dm.zip](#) < 3dm Rhinoceros file

**Figure 96 : Capture d'écran de la description du sample Panels-dispatch, source : [co-de-it]**

## **7.2 Spécificités cognitives de l'usage de Samples pour la conception de modèles paramétriques en conception architecturale**

Pour étudier les activités cognitives impliquées par l'usage de samples, nous nous intéressons dans un premier temps à l'activité d'exploration des samples disponibles et de choix de samples spécifiques (7.2.1) puis, dans un deuxième temps, nous nous intéressons aux opérations de conception de modèles paramétriques impliquées lorsque des samples sont utilisés (7.2.2).

### **7.2.1 Exploration de bibliothèques et choix d'un sample : prise de décision et attribution de mesure à un modèle paramétrique**

L'analyse des vidéos de captures d'écran (*cf.* annexe 2 p. 406) montre plusieurs comportements récurrents liés à l'exploration de samples en vue de leurs usages. Lors des expérimentations, cette exploration s'est appuyée sur différents supports :

- Les catalogues de samples du cours et des sites référencés dans le cours;
- Les supports de cours, souvent explorés en parallèle des catalogues;
- Internet et les moteurs de recherche, pour expliciter des notions. C'est ainsi que le binôme F1 (Expe 1, *cf.* annexe 2 p. 406) s'appuie sur Google pour chercher des informations sur les diagrammes de Voronoi ou encore explore des fonctions permettant de générer des courbes paramétriques sur le site [mathcurve.fr].

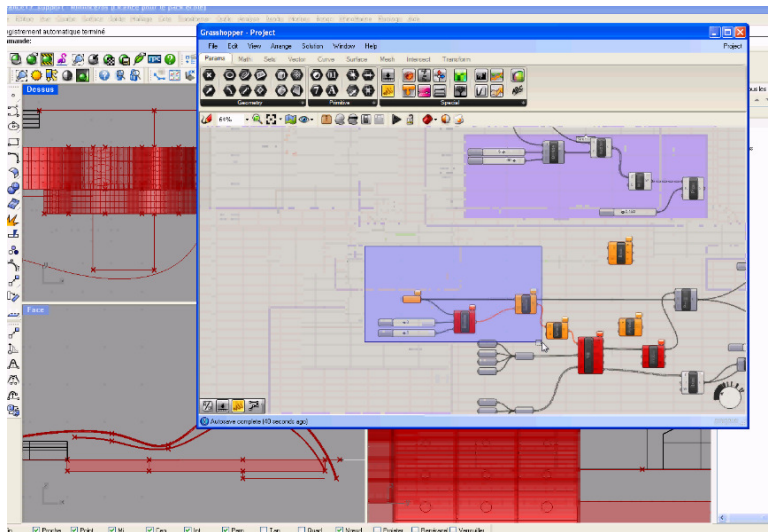
Les samples eux-mêmes ne sont donc pas les uniques sources permettant leurs usages. Deux autres types de supports sont explorés : les cours suivis par l'étudiant et les ressources du web. Par ces explorations, l'étudiant interroge des savoirs géométriques ou informatiques qui lui permettent de s'approprier les samples. En effet, l'appropriation d'un fragment de modèle développé par un autre fait appelle à des références, techniques ou culturelles (*cf.* Chapitre 3) qui peuvent être nouvelles pour l'étudiant et qu'il lui faut donc construire.

Ces recherches d'informations sont complétées par une exploration du contenu des samples eux-mêmes. Le Sample choisi est intégré dans le modèle : -par copier-coller d'un fichier existant (personnel ou non) (*cf.* groupes A3, B3, C2,...) ; ou -par reconstruction d'une partie ou de la totalité d'un Sample (*cf.* groupes A2 et B3). Dans le cas où le sample est copié-collé, les entrées et sorties du modèle peuvent être testées et manipulées directement. L'exploration du sample se fait alors par l'observation et le test des impacts de modifications successives. Dans ce cas, l'exploration s'accompagne déjà d'une appropriation : le sample, est testé pour être compris, puis modifié en fonction de la nouvelle modélisation visée. Lors de ces explorations, les étudiants font parfois des transformations majeures du modèle et, ont parfois besoin de revenir au fichier original, en l'ouvrant ou en observant les images qui rendent comptes du modèle d'origine (images issues des supports de cours pour les cas des Exp2 et 3).

Finalement, on observe deux types d'exploration des samples:

- **Exploration des samples au travers de descriptifs disponibles** (sites internet...): L'exploration se fait par l'observation des images et des textes de description. Dans ce cas, les supports de cours auxquels se rapportent les samples sont tout particulièrement consultés. Des liens directs entre les samples et des notions permettent aux concepteurs de construire la culture nécessaire à leur compréhension et leur appropriation des patterns. Cette exploration participe à la construction d'*espaces de référence* nécessaires à la conception des modèles paramétriques visés.

- **Exploration par le test du contenu du sample** (cf. Figure 97) : Elle se fait par des modifications successives et des tests d'adaptation du sample au modèle du projet. Dans ce cas, la lisibilité du contenu du sample doit être tout particulièrement soignée.



**Figure 97 : Exploration d'un sample par son contenu, copier-coller dans le modèle paramétrique de l'étudiant, Capture d'écran du binôme H3 de l'Expe2**

La spécification de ces deux mécanismes d'exploration et de compréhension des samples permet de proposer, dès à présent, des éléments nécessaires à associer à une aide à la modélisation paramétrique par patterns appliqués :

- les notions techniques (c'est-à-dire les savoirs, comme les structures de données, les produits vectoriels et la mise en œuvre de ces savoirs comme « créer une liste de liste » « faire le produit de deux vecteurs ») se rapportant aux samples doivent être explicitées,
- les contenus des modèles doivent être visibles au premier coup d'œil, sans avoir besoin d'ouvrir le sample.

## **7.2.2 Stratégies d'appropriation de samples : *dimensionner* un modèle paramétrique conçu par un autre**

Dans le protocole de l'expérimentation, il était demandé aux étudiants d'utiliser des samples (ou patterns appliqués) pour modéliser un projet architectural qui leur serait propre. Pour cela, les étudiants se sont appropriés des fragments de modèles paramétriques conçus par d'autres, en vue d'accompagner leur processus de conception architecturale. Deux stratégies d'appropriation peuvent être identifiées :

- l'appropriation par **réutilisation** de la solution technique proposée par le sample ;
- l'appropriation par **reconstruction** d'une solution technique, à partir de l'exemple proposé par le sample.

### 7.2.2.1 Réutilisation de la solution technique proposée par le sample

Dans la plupart des cas d'usage de samples observés lors des expérimentations, le sample utilisé par les étudiants était copié-collé au sein du modèle développé, puis manipulé et transformé pour correspondre à leurs intentions. Dans ces cas là, le fragment de modèle proposé par le sample est directement intégré au modèle paramétrique conçu par les étudiants, puis il fait l'objet d'opération de conception visant à lui donner des mesures spécifiques au modèle paramétrique visé. Le sample est alors utilisé comme une dimension (ce sur quoi porte la mesure) d'opérations de conception de modèle paramétrique (cf. Figure 98).

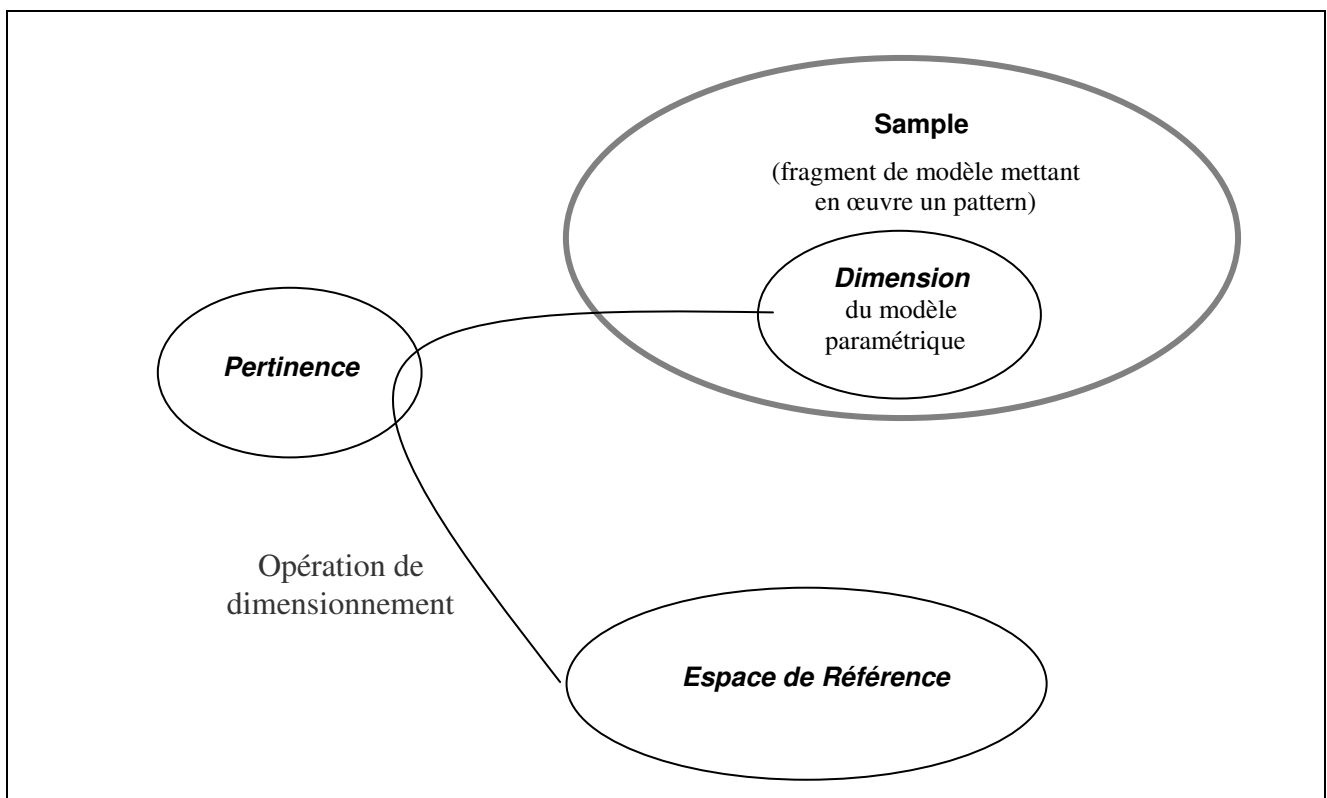


Figure 98 : Utilisation d'un sample dans une opération de dimensionnement directement en tant que fragment de modèle sur lequel porte la mesure

Lors de la manipulation directe des samples, trois modes de dimensionnement peuvent être observés :

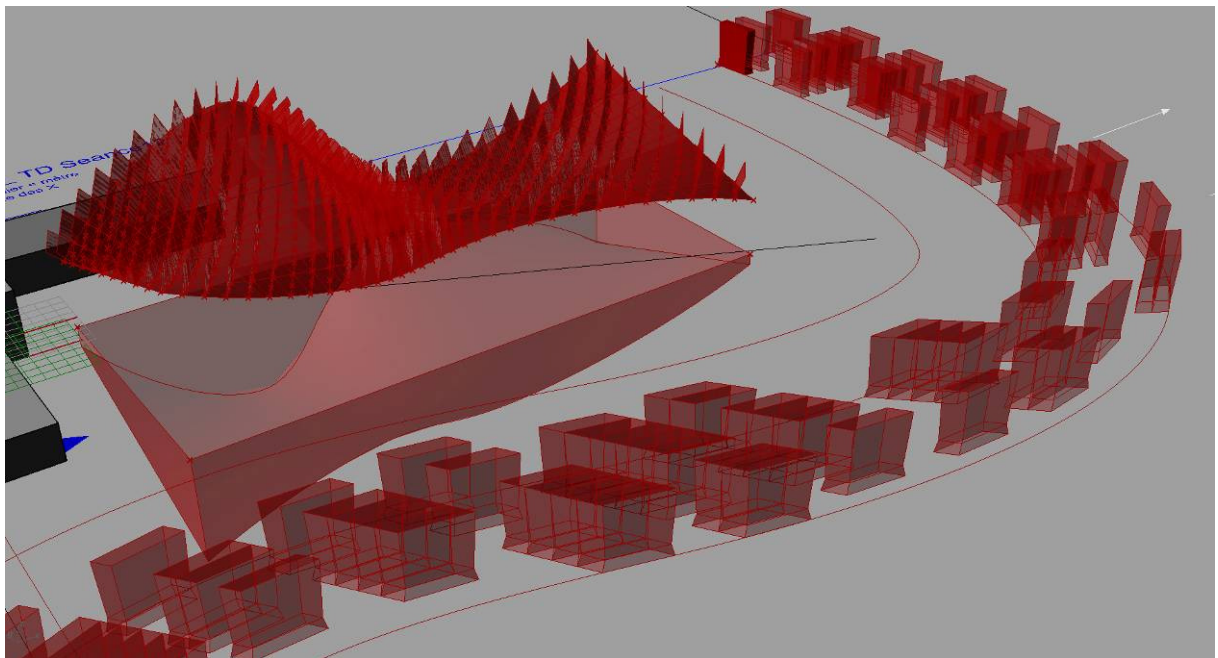
- le dimensionnement des variables numériques indépendantes du sample ;
- le dimensionnement d'entrées spécifiques au sample ;
- la transformation directe de la chaîne de dépendance établie dans le sample.

Ces différentes stratégies de *dimensionnement* sont décrites dans les paragraphes suivants.

### - Dimensionnement de variables numériques

Dimensionner les variables numériques indépendantes (les paramètres) d'un sample en fonction d'une *pertinence* du modèle paramétrique visé est un mode courant et souvent incontournable de l'appropriation d'un sample. Cette manipulation de variables numériques est simple (déplacer un slider par exemple) et plus accessible qu'une réelle transformation du modèle préconçu, demandant des compétences plus avancées en modélisation.

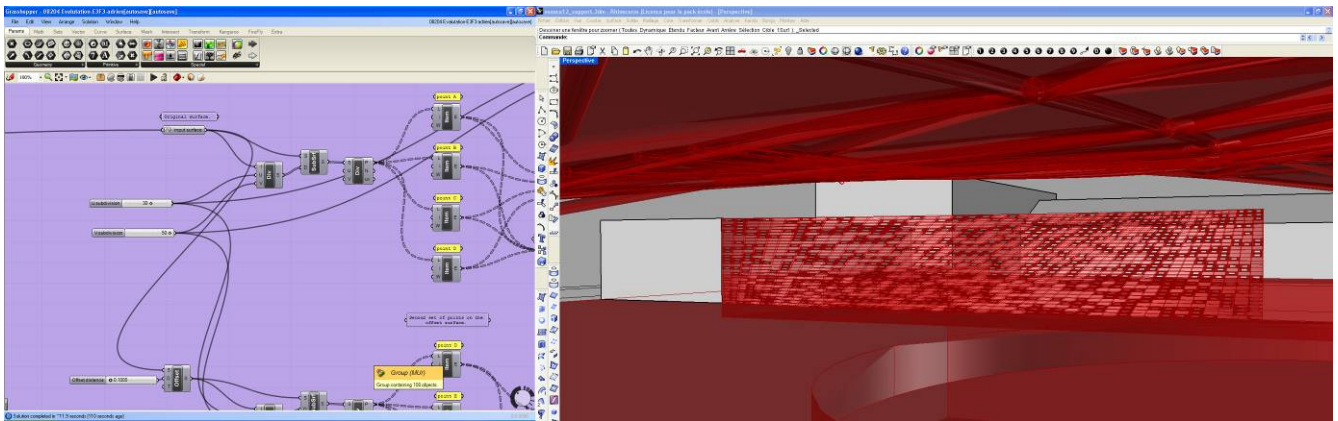
Par exemple, le binôme A3 (Expe 2) s'est approprié le sample Panel-dispatch en dimensionnant le paramètre « hauteur » d'un panneau en fonction d'une *pertinence* fonctionnelle : les volumes des « panneaux » étant utilisés pour modéliser des cabines de bain (cf. Figure 99). Ici la manipulation du paramètre hauteur<sup>88</sup> met en œuvre une nouvelle opération de mesure du modèle paramétrique, transformant la génération de panneaux à claire-voie attribuée au sample par ses concepteurs (l'équipe de Co-de-it [co-de-it]), en génération de cabine de bain, voulue par les étudiants.



**Figure 99 : Dimensionnement du sample Panels-dispatch en cabines de bain par le binôme A3 (EXPso921-2011)**

<sup>88</sup> Cette manipulation de variables est, dans le cas de ce projet, en plus associée à une manipulation du contenu du modèle. Celle-ci est décrite dans le paragraphe suivant.

Le binôme E3 a, quant à lui, *dimensionné* le sample Panels-dispatch en adaptant les paramètres définissant la densité des panneaux (cf. Figure 100). Ces paramètres ont été définies selon la *visibilité*<sup>89</sup>, l'*optique*<sup>90</sup> et le *modèle*<sup>91</sup> (Boudon et al. 2000, p.174-177) en vue d'obtenir un « moucharabieh » tel que les concepteurs le décrivent dans leur note d'intention architecturale : « *Le mur fonctionne tel un moucharabieh (échelle de modèle) pour masquer les baigneurs (échelle fonctionnelle) tout en laissant entrevoir la vue vers l'ouest (échelle de visibilité). Le pattern nous a permis de réaliser la transition architecturale souhaitée (échelle technique, échelle de représentation).* » (Note d'intentions architecturales, binôme E3, Expe2).



**Figure 100: Dimensionnement du sample Panel-dispatch par le groupe E3 de l'Expe2**

### - Conception d'entrées spécifiques

La conception d'« entrées » spécifiques est une autre des stratégies de *dimensionnement* des samples observés. L'« entrée » d'un modèle paramétrique est une donnée ou un groupe de données visant à être traité.

Par exemple, dans le cas du projet du binôme C1 (cf. Annexe 2.16 p. 389), la conception d'une surface est utilisée comme l'entrée de deux samples : le sample Panel\_Dispatch générant une couverture aux ouvertures aléatoires et le sample Créer-motif-maillage générant une structure en treillis sur la partie inférieure de la couverture (cf. Figure 101). Les concepteurs écrivent à ce sujet : « *L'enveloppe de la piscine principale serait composée de deux sous-enveloppes : l'une ayant une fonction structurelle, l'autre une fonction esthétique tel un signal dans le paysage* » (Note d'intentions architecturales, binôme C1, Expe1). Cette assertion met en avant la pertinence *technique*<sup>92</sup> donnée au sample Créer-motif-maillage, et la pertinence *optique*<sup>93</sup> donnée au sample Panel-dispatch. Ici, pour l'utilisation de ces

<sup>89</sup> L'échelle de visibilité rend compte d'opérations cognitives visant à « *situer un objet ou une partie d'objet de manière à ce qu'il soit vu d'un lieu ou qu'il ait vue sur un lieu* » (Boudon et al. 2000, p.174).

<sup>90</sup> L'échelle optique définit les opérations mentales de « *tenir compte d'un point de vue –au sens propre- pour donner à une partie ou au tout de l'espace architectural une mesure faisant intervenir les modalités suivant lesquelles elle va être vue* » (Boudon et al. 2000, p.175).

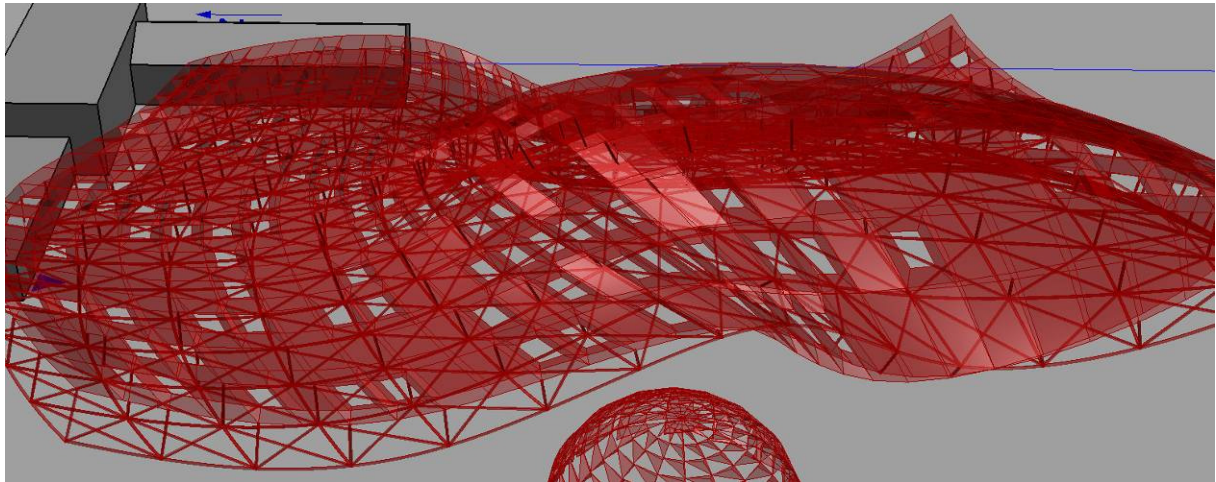
<sup>91</sup> L'échelle de modèle désigne les opérations cognitives visant à « *repréparer un modèle antérieur, tout en effectuant éventuellement des modifications de divers degrés et de diverses natures* » (Boudon et al. 2000, p.177).

<sup>92</sup> L'échelle technique correspond à l'activité de « *Utiliser des considérations d'ordre technique pour induire une modalité d'attribution de mesures à une partie ou un tout de l'espace architectural* » (Boudon et al. 2000, p.175).

<sup>93</sup> L'échelle optique correspond à l'activité de « *tenir compte d'un point de vue –au sens propre- pour donner à une partie ou à un tout de l'espace architectural une mesure qui fasse intervenir les modalités suivant lesquelles il est vu* » (Boudon et al. 2000, p.167).



deux samples dans le modèle paramétrique voulu, une surface a été conçue sur Rhinocéros selon d'autres *pertinences*. Cette surface répond, entre autres, à des *échelles de voisinage*<sup>94</sup> (la surface se soulève au niveau des bâtiments existants) et de *visibilité*<sup>95</sup> (la surface est orientée vers la « vue dégagée »).



**Figure 101: Capture d'écran du modèle paramétrique développé par le binôme C1 (Expe 1)**

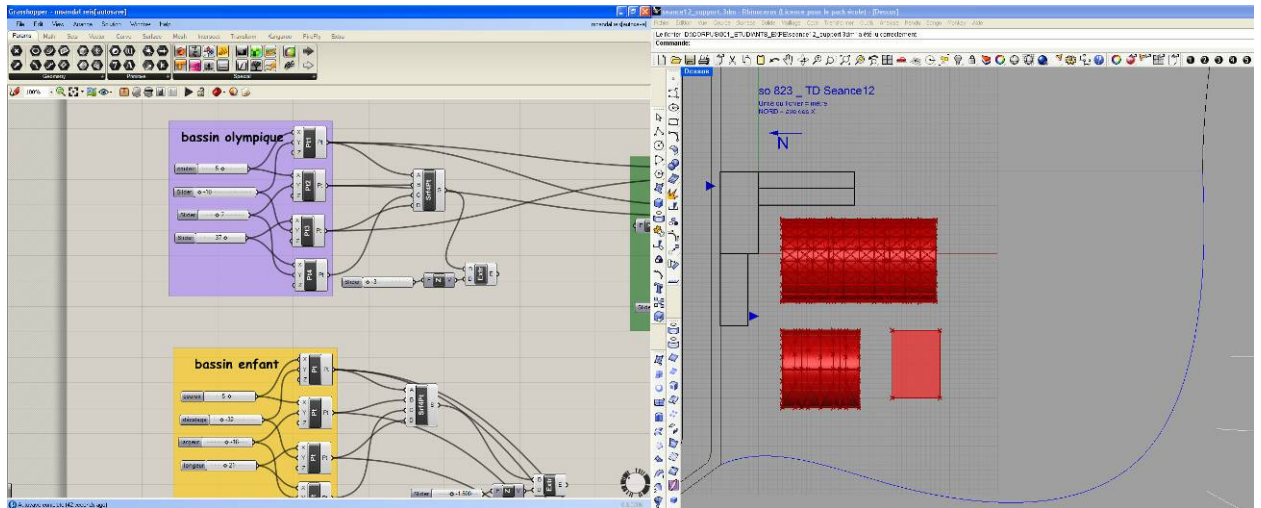
La conception d'entrées spécifiques aux samples a parfois été, pour les étudiants, le lieu d'opérations de conception de modèles paramétriques visant à positionner les géométries générées dans le site imposé. Plusieurs stratégies ont été mises en place par les étudiants pour ancrer leur modèle paramétrique modélisé sur Grasshopper dans le site modélisé sur Rhinocéros (sur les relations entre Grasshopper et Rhinocéros, cf. 2.3.1). On observe par exemple des stratégies visant à :

- Placer des points sur Grasshopper dont les coordonnées renvoient au système de Rhinocéros ;
- Faire appel dans Grasshopper à des éléments modélisés dans la scène Rhinocéros : comme des éléments du site (la surface de la parcelle par exemple) ou des éléments créés par les étudiants (même si cela leur a été interdit dans les protocoles des expérimentations).

---

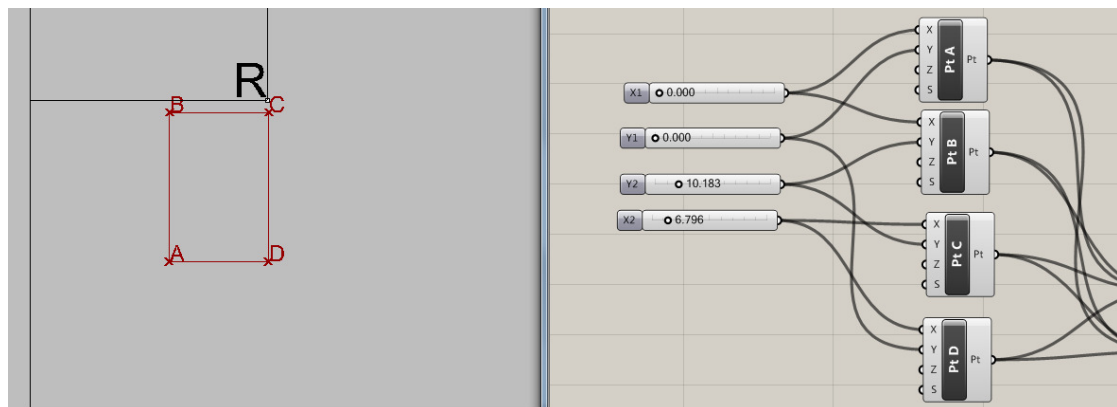
<sup>94</sup> L'échelle de voisinage consiste à : « attribuer des mesures par contiguïté, relativement à des éléments appartenant au voisinage et induisant une continuité spatiale » (Boudon et al. 2000, p.171) .

<sup>95</sup> L'échelle de visibilité rend compte d'opérations cognitives visant à « situer un objet ou une partie d'objet de manière à ce qu'il soit vu d'un lieu ou qu'il ait vue sur un lieu » (Boudon et al. 2000, p.174).



**Figure 102 : Dimensionnement de surfaces à partir de coordonnées absolues par le groupe C2 (EXPso823-2011)**

Ainsi certains étudiants ont placé des points sur Grasshopper en définissant leurs coordonnées avec des valeurs définies intuitivement sur Rhinocéros et en lien avec les dimensions de l'objet créé. Par exemple, le groupe C2 (cf. Figure 102) a modélisé des bassins à partir de quatre points dont les coordonnées sont liées à l'implantation des piscines dans le site, ainsi qu'à leurs longueurs et largeurs, elles-mêmes liées aux fonctions des piscines (pataugeoire, piscine olympique et bassin pour enfants). Cette définition du modèle est difficilement modifiable. En effet elle nécessite de nombreuses variables numériques indépendantes dont les pertinences ne sont pas forcément explicitées. Par exemple dans le modèle de la Figure 103 le rectangle ABCD est modélisé de façon à être « paramétré » par les coordonnées absolues de ses sommets.

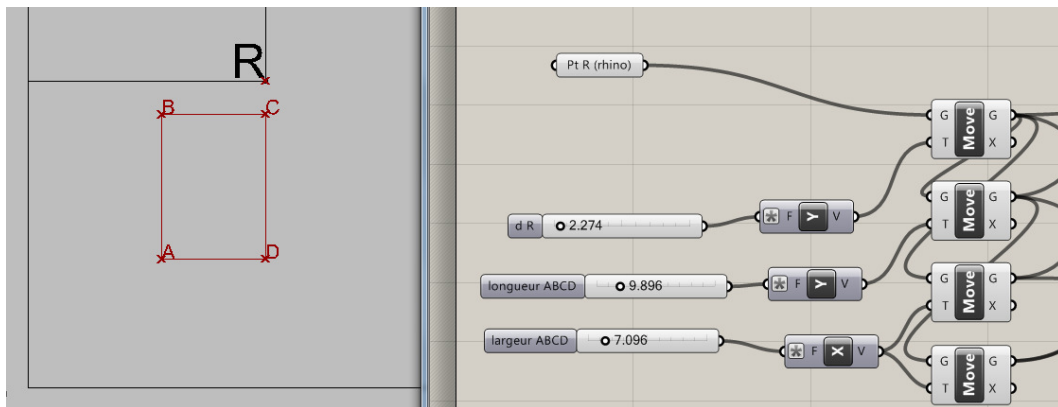


**Figure 103: Exemple de modélisation d'un rectangle par les coordonnées en valeurs absolues de ses sommets**

Les transformations de ce rectangle sont grossières : toutes sont possibles, mais elles sont malaisées car il est difficile de pondérer des valeurs absolues. Par exemple si on souhaite transformer mon rectangle en fonction de sa distance avec le point R (c'est-à-dire en rapport avec une échelle de voisinage), il vaut mieux avoir explicité le rapport du rectangle ABCD à R lors de la modélisation,

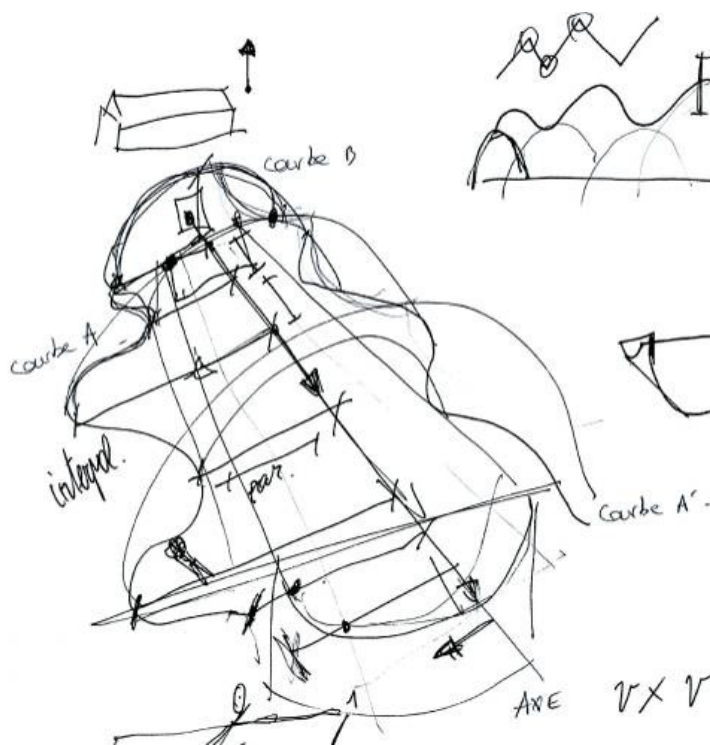


comme dans le modèle proposé dans la Figure 104. Dans le modèle de la Figure 104, le rectangle ABCD est défini par sa longueur, sa largeur et sa distance minimale au point R.



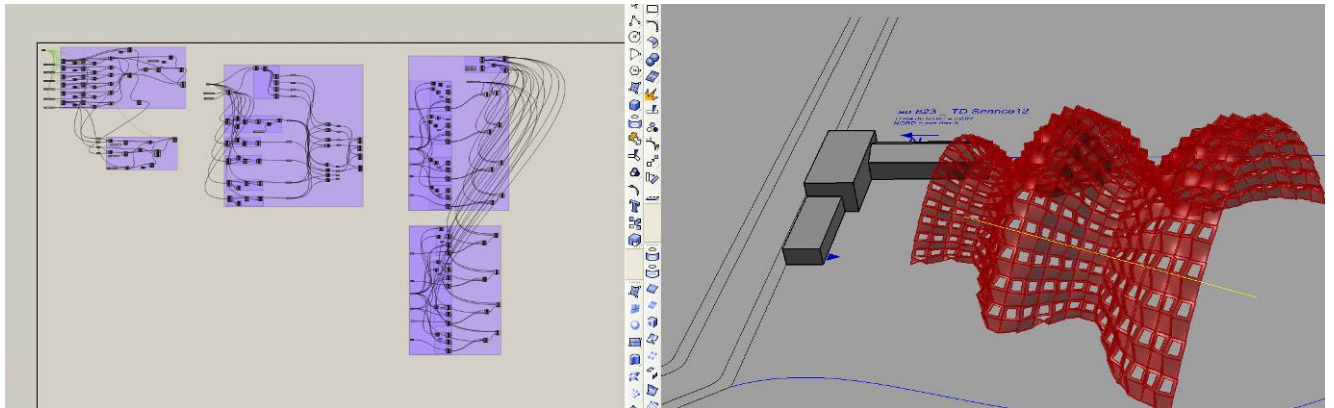
**Figure 104 : Exemple de modélisation d'un rectangle ABCD en fonction de sa longueur, de sa largeur et de sa distance au point R**

Comme dans le cas de l'exemple proposé dans la Figure 104, la plupart des étudiants se sont appuyés sur des éléments de la scène Rhinocéros pour positionner les géométries produites par leurs modèles paramétriques. Ces éléments peuvent être les géométries du site (contour de la parcelle, points remarquables, *etc.*) ou être construits, par les étudiants eux-mêmes, par commodité. C'est le cas du groupe A2 qui a décidé de penser son projet à partir d'un axe « *vers la vue dégagée* »<sup>96</sup> (cf. Figure 105). Par facilité, cet axe a été dessiné sur Rhinocéros puis directement utilisé dans le modèle Grasshopper (cf. Figure 106).



<sup>96</sup> Extrait de retranscription, groupe A2, 9h48 : Sebastiaan : « on peut commencer par définir l'axe », Rebecca : « vers la vue dégagée », Sebastiaan « l'axe sur rhino et les points on peut les définir comme ça » (cf Annexe p. 396).

**Figure 105 : Esquisses produites par le groupe A2 lors de l'Expe 1**

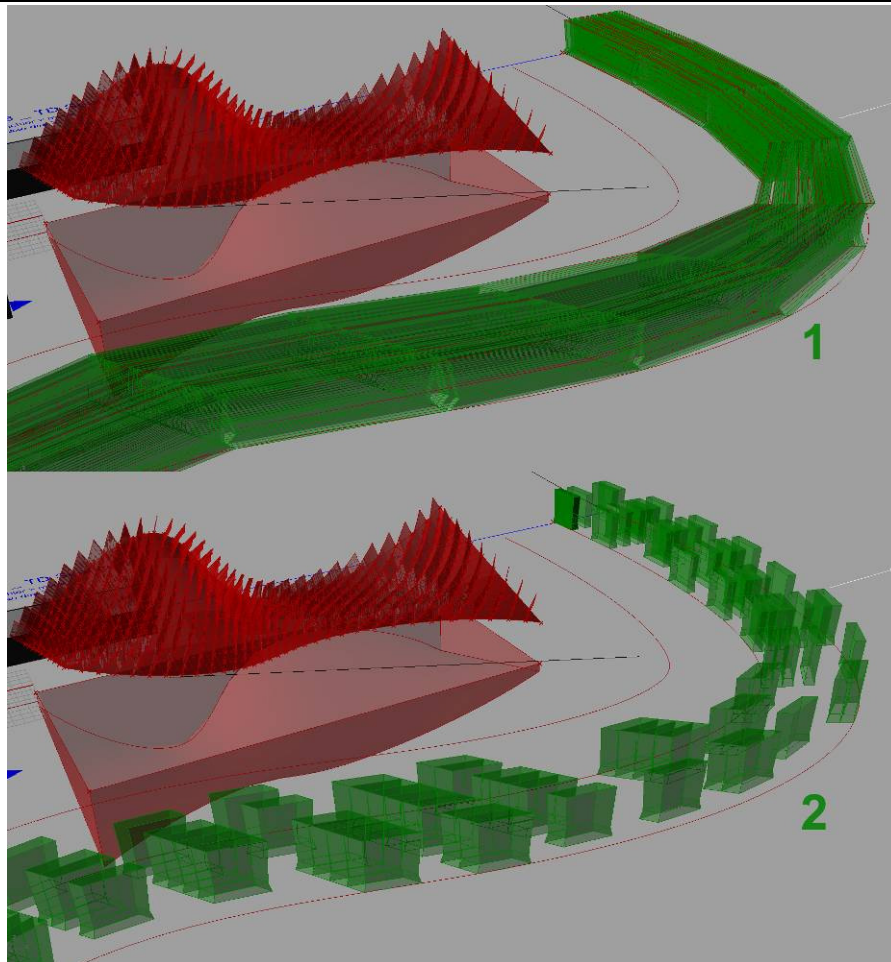


**Figure 106 : Modèle construit par le groupe A2 (Expe 1)**

### **- Dimensionnement de la chaîne de dépendance établie dans le sample**

Sur les onze samples dont l'appropriation a été étudiée (cf. Annexe 3), neuf ont été « copiés-collés » en ne faisant l'objet que de modifications de données d'entrées et de variables (cf. paragraphes précédents). De rares cas ont procédé à des opérations de dimensionnement de la chaîne de dépendance définie par le sample lui-même. Ces appropriations de samples donnent lieu à une modélisation riche mais restent peu courantes.

Nous avons déjà pris pour exemple précédemment le cas d'appropriation par le binôme A3 du sample Panel-dispatch pour générer des cabines de bain à partir d'un sample initialement conçu par [co-de-it] pour générer des calepinage de façade. Nous avons vu que les paramètres numériques du sample ont été modifiés. La variable de hauteur des panneaux a été mise à une « échelle fonctionnelle » des « cabines de bain ». Le sample a également été modifié pour transformer l'implantation des panneaux-cabines. Par défaut, le sample Panel-dispatch découpe une surface d'entrée en sous-surfaces sur lesquelles sont générées aléatoirement des panneaux. Afin de ménager des accès aux cabines de bains (*échelle fonctionnelle* inscrit dans l'*espace de conception architecturale*) le modèle paramétrique du sample a été dimensionné pour que les sous-surfaces générées, sur lesquelles se construisent aléatoirement les panneaux-cabines, soient réduites à une rangée sur deux (cf. Figure 107). Ce *dimensionnement* consiste en la modification de la structure du flux de données parcourant le sample, en vue de ne conserver qu'une rangée sur deux de sous-surfaces (cf. Figure 107).



Légende

1 : test du sample Panel-dispatch d'origine avec les entrées et les variables définies par les étudiants

2 : le sample Panel-dispatch tel que modifié par les étudiants

**Figure 107 : Appropriation du sample Panel-dispatch par le groupe A3**

On observe donc ici que lorsqu'un sample est utilisé en étant directement l'objet d'opération de conception de modèle paramétrique en vue de concevoir le modèle visé, deux types de *découpage*<sup>97</sup> ont été observés. On observe des opérations de *dimensionnement* portant sur les variables indépendantes des samples, que ces variables soient numériques ou géométriques, et des opérations de *dimensionnement* portant sur la chaîne de dépendances du sample (cf. Figure 108).

<sup>97</sup> L'opération de *découpage* est celle par laquelle ce sur quoi porte la mesure, c'est-à-dire la *dimension*, est identifié, cf. 6.1.1

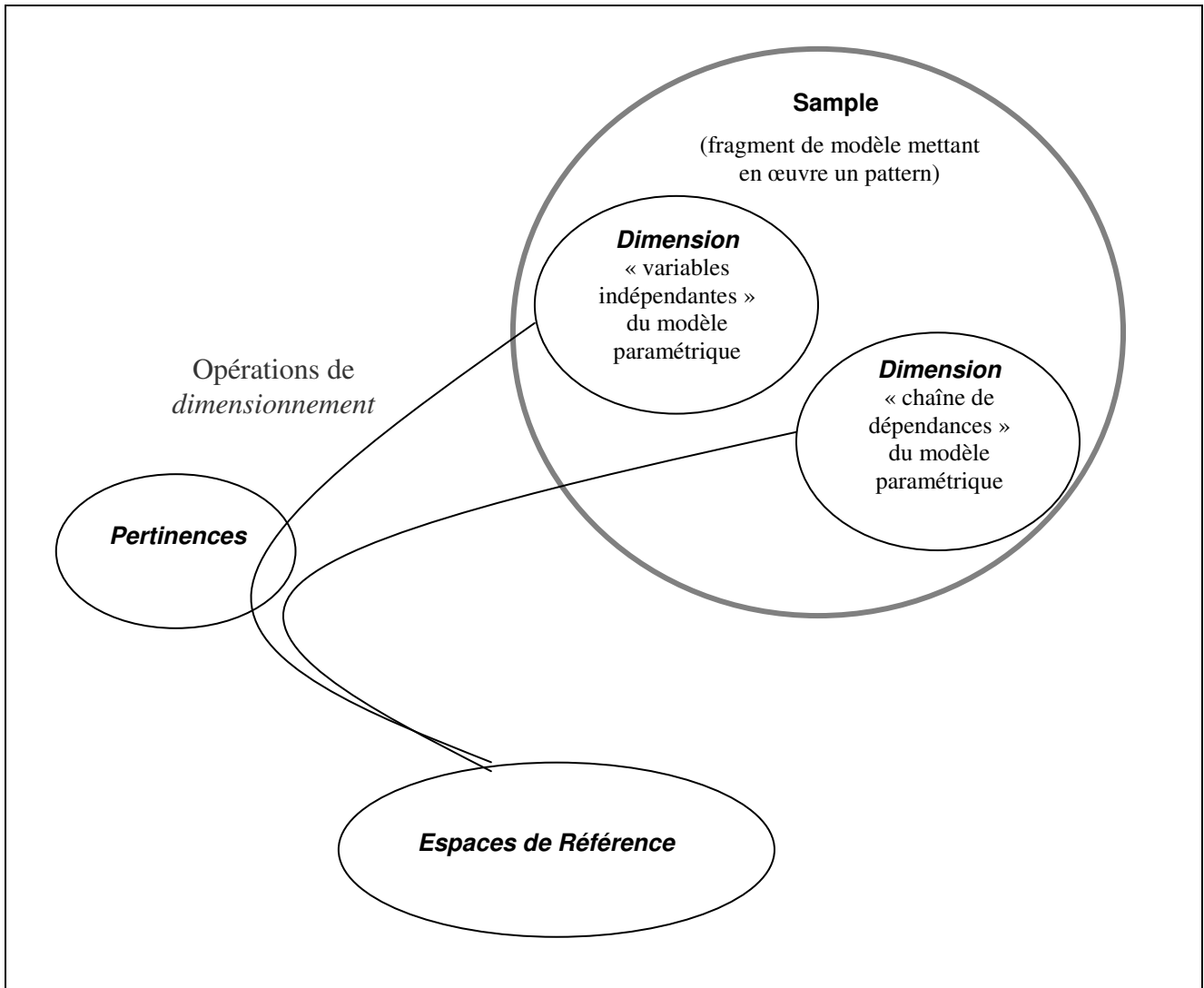
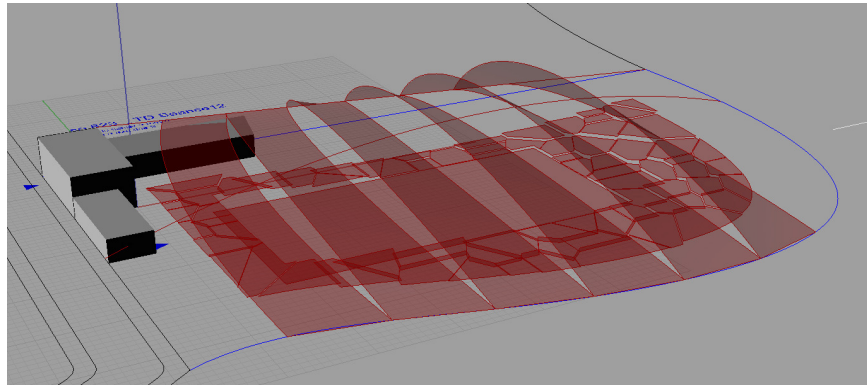


Figure 108 : identification de dimensions possibles lors d'opérations de dimensionnements d'un sample

### 7.2.2.2 Reconstruction de la solution technique proposée par le sample

Nous avons analysé dans la section précédente des cas de *dimensionnement* direct de samples dans le cadre de conception de modèles paramétriques en conception architecturale. Or, on peut également observer des cas où les samples sont utilisés sans être directement intégrés dans le modèle paramétrique en cours de conception.

Par exemple le binôme D1 (cf. Figure 109) a commencé par copier-coller les samples *Srf\_Projetée* et *Crv\_Voronoi* depuis la bibliothèque du cours, pour ensuite les manipuler et recombina certains composants en vue de construire un code très différent. Du sample *Srf\_Projetée* a été repris un fragment de code permettant de générer des points aléatoires. Du sample *Crv\_Voronoi* a été repris l'usage du composant générant un diagramme de Voronoi à partir d'un nuage de points. Les deux samples semblent ici avoir servis d'exemples d'usage de composants, c'est-à-dire de support pour un apprentissage autonome de fonctionnalités inconnues des étudiants.



**Figure 109 : Capture d'écran du projet développé par le binôme D1 (Expe 1)**

Dans ce cas, les samples ont été utilisés par les étudiants comme une aide à la construction d'un *espace des savoirs de la modélisation paramétrique*, tel que défini dans la section 6.4.2.2 (cf. p. 223) comme espace théorique des savoirs, savoir-faire et compétence participant à la conception de modèles paramétrique en conception architecturale. Dans cet exemple, l'usage des samples a permis aux étudiants de se construire des références et des points de vue grâce auxquels ils ont donné de la mesure à leur modèle paramétrique. Les données dont nous disposons ne permettent pas de savoir si ces références et ces points de vue ont également participés à la conception architecturale ou seulement à des opérations de *traduction en géométrie paramétrique* (cf. Figure 110).

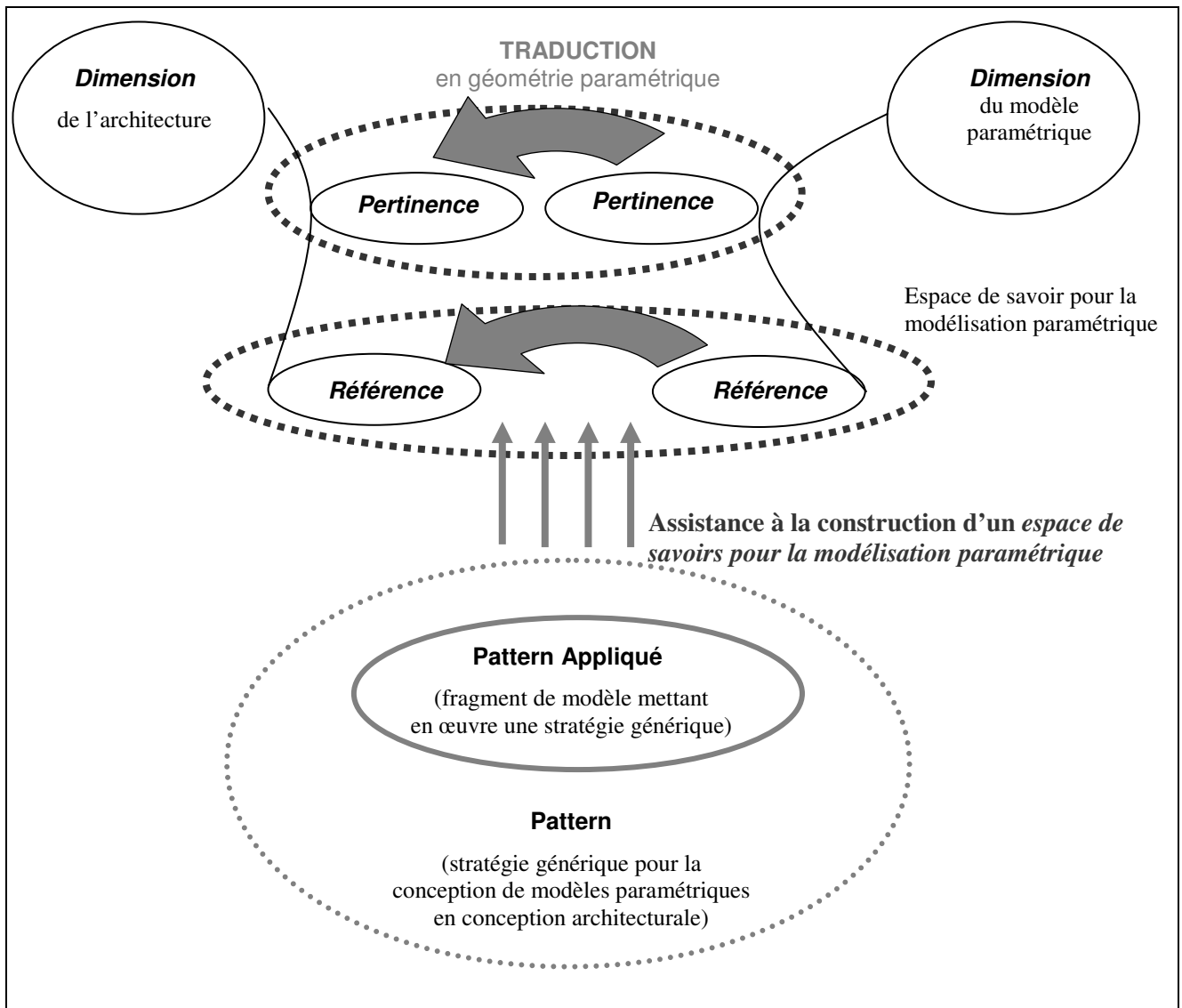


Figure 110 : Usage d'un sample comme assistance à la conception de modèle paramétrique en conception architecturale

### **7.3 Discussion de l'usage des patterns et des samples comme aide à la modélisation paramétrique pour la conception architecturale**

L'analyse des expérimentations menées montre que le recours aux samples est une aide à la modélisation paramétrique pour la conception architecturale. Les samples permettent tout particulièrement :

- d'assister la constitution de modèles paramétriques grâce aux catalogues de modèles « clefs en main » ;

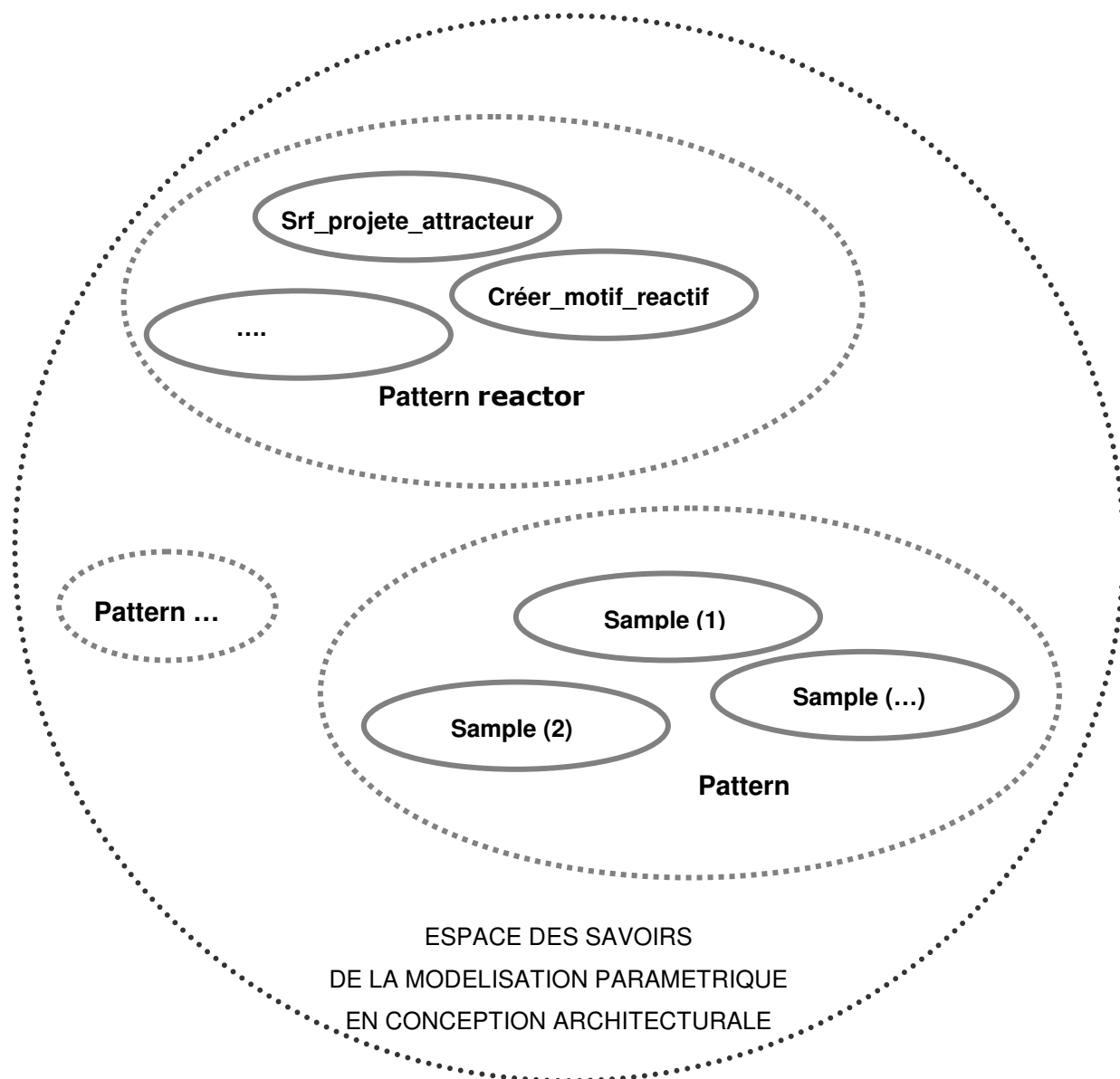
- d'assister la mise en œuvre au sein du processus de l'*échelle de modèle* de conception du modèle paramétrique par la possibilité de découvrir et/ou de reprendre des exemples et des idées de résolution de questions techniques ;

- et enfin ils permettent aux étudiants d'apprendre le fonctionnement de composants ou de technique de modélisation par eux-mêmes grâce à l'exemplification.

Malgré tout, les difficultés de modélisation avec les samples sont présentes. L'étudiant peut ne pas savoir ce qu'il cherche (*cf.* binôme F1), ne pas trouver un sample adapté à sa question technique ou ne pas réussir à s'approprier un sample (*cf.* binôme A3).

Ainsi, lors des expérimentations, nous avons analysé l'usage de patterns appliqués pour la conception de modèles paramétriques en conception architecturale. Les patterns proposés par Robert Woodbury se sont révélés difficiles à communiquer à des débutants en modélisation paramétrique. C'est l'application des patterns dans des cas particuliers : les samples, qui a permis de les rendre accessibles. L'application des patterns dans des samples rend effectif l'aide à la modélisation paramétrique de Woodbury en créant une médiation entre les stratégies très génériques des patterns et les cas particuliers de modélisation impliqué lors de la conception d'un modèle paramétrique en conception architecturale.

Nous avons analysé cela au travers de l'usage des samples comme assistance à la construction d'un *espace des savoirs de la modélisation paramétrique en conception architecturale* (*cf.* Figure 110). En appliquant des patterns à des questions techniques particulières, les samples exemplifient des stratégies et des solutions techniques. A partir de ces différentes exemplifications de stratégies et de questions techniques récurrentes et spécifiques à la conception de modèles paramétriques pour la conception architecturale, le concepteur peut, par induction, se constituer des savoirs plus généraux relatifs à la modélisation paramétrique en conception architecturale (*cf.* Figure 111).



**Figure 111 : Participation des patterns et des samples à un *espace des savoirs de la modélisation paramétrique***

Pour être efficaces, les samples ne peuvent donc pas être isolés des notions techniques et/ou culturelles qui les sous-tendent. Hors des expérimentations, les recours aux supports de cours ont été majeurs. La construction d'un *espace de savoirs de la modélisation paramétrique en conception architecturale* est indispensable aux usages de ces samples.





## **8 Proposition d'une pédagogie de la modélisation paramétrique en conception architecturale**

Notre recherche nous a permis de produire des résultats sur les opérations cognitives impliquées lors d'usages de la modélisation paramétrique (cf. Chapitre 6) et lors d'usages des patterns en architecture (cf. Chapitre 7). Ces résultats nous amènent à proposer une pédagogie de la modélisation paramétrique en conception architecturale.

Dans un premier temps, ce chapitre présente le contexte et le positionnement de la pédagogie proposée (8.1.1), puis expose un programme pédagogique au travers : d'objectifs pédagogiques fixés (8.1.2), de supports didactiques (8.2) et de scénarii pédagogiques (8.3).

### **Plan du chapitre :**

|   |            |
|---|------------|
| <b>8.1 Objectifs pédagogiques de l'apprentissage de la modélisation paramétrique par des architectes.....</b>                   | <b>261</b> |
| 8.1.1 Contexte et positionnement de la pédagogie proposée   | 261        |
| 8.1.2 Construction d'objectifs pédagogiques pour l'apprentissage de la modélisation paramétrique par des architectes            | 262        |
| <b>8.2 Supports didactiques pour l'apprentissage de la modélisation paramétrique en conception architecturale .....</b>         | <b>274</b> |
| 8.2.1 [parametric-ressources] : publication en ligne de samples   | 275        |
| 8.2.2 Support didactiques : [dnarchi] Plateforme de publication de savoirs  | 283        |
| <b>8.3 Support pédagogique, pour un enseignement de la modélisation paramétrique en conception architecturale .....</b>         | <b>287</b> |
| 8.3.1 Inscription d'un enseignement de la modélisation paramétrique dans les cursus d'enseignements de l'architecture existants | 287        |
| 8.3.2 Pédagogie par Travaux Dirigés   | 289        |
| 8.3.3 Pédagogie par projet  | 302        |

## 8.1 Objectifs pédagogiques de l'apprentissage de la modélisation paramétrique par des architectes

### 8.1.1 Contexte et positionnement de la pédagogie proposée

Dans la présente recherche, nous entendons par « pédagogie » le champ s'intéressant à la relation entre formateur et apprenant, tel que le proposent Altet (2006) et Best (1994). Pour ces auteurs, la pédagogie est « une réflexion sur les finalités de l'éducation et une analyse objective de ses conditions d'existence et de fonctionnement » (Best in Champy et al. 1994, p.727). Plus précisément, Marguerite Altet, dans *Les pédagogies de l'apprentissage* (2006), définit la pédagogie comme l'articulation entre des processus d'enseignement et d'apprentissage, mettant en jeu des savoirs et des finalités (cf. Figure 112). L'enseignement désigne un processus de transmission de savoirs par un enseignant. L'apprentissage, quant à lui, désigne le processus d'acquisition de connaissances par le faire (au sens courant). La pédagogie est alors l'articulation entre ces deux processus (transmission de savoir par l'enseignant et acquisition de ce savoir-faire par l'apprenant) (Altet 2006, p.16).

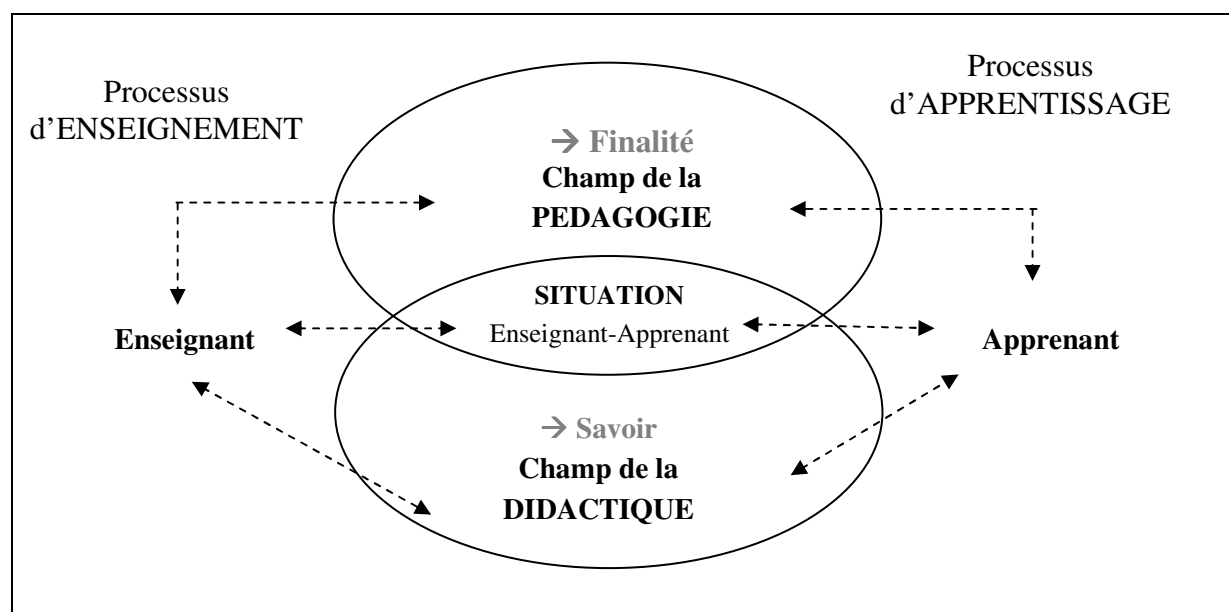


Figure 112: Articulations des processus d'enseignement et d'apprentissage d'après (Altet 2006, p.17)

Le modèle de définition de la pédagogie d'Altet (cf. Figure 112) met l'accent sur la situation d'apprentissage et les échanges entre l'enseignant et l'apprenant. La pondération et la nature des échanges permettent de qualifier les différentes pédagogies possibles. Ainsi, une pédagogie centrée sur la prestation de l'enseignant et les savoirs constitués qu'il transmet peut être nommée « magistro-centriste » (Altet 2006, p.17). A l'inverse, une pédagogie centrée sur l'apprenant et ses processus d'apprentissage sera nommée « pédagogie de l'apprentissage » (Altet 2006, p.20).

Comment définir le positionnement pédagogique le plus pertinent pour la modélisation paramétrique en conception architecturale ? Ce sont les particularités de la pratique de la modélisation paramétrique qui nous aide à définir ce positionnement.

Si certains savoirs sollicités par l'usage de la modélisation paramétrique en architecture sont fondamentaux et stables, comme les savoirs géométriques ou informatiques fondamentaux par exemple, d'autres sont éminemment « périssables ». Ainsi les modeleurs paramétriques et leurs extensions (les plugins comme Kangaroo ou GHowl pour Grasshopper, les programmes complémentaires, *etc.*) évoluent actuellement très rapidement<sup>98</sup>. Ces mises à jour demandent des explorations et des recherches d'informations sur les nouvelles fonctionnalités disponibles<sup>99</sup>. La modélisation paramétrique est une pratique mouvante, reflétant les innovations et les évolutions du paysage des pratiques architecturales comme des avancées technologiques. Les contenus d'un tel enseignement doivent donc être à jour. Sans vouloir établir une course à la nouveauté ou à l'innovation pour l'innovation, les contextes professionnels auxquels les étudiants seront destinés évoluent rapidement, surtout sous l'angle de l'usage des technologies numériques. Si un individu veut rester compétant dans l'usage d'un logiciel, il doit être capable de trouver rapidement l'information qui lui manque sur une mise à jour, une incompatibilité de version, *etc.* Il y a donc ici un fort enjeu dans **l'apprentissage** de la modélisation paramétrique: c'est-à-dire dans le processus d'assimilation de la modélisation paramétrique en architecture par l'apprenant. Un enseignement de la modélisation paramétrique doit prendre en compte la nécessité première de former les étudiants à apprendre, pour ensuite leur permettre d'apprendre par eux-mêmes.

Une pédagogie de la modélisation paramétrique pour les architectes doit donc viser à transmettre :

- des bases d'apprentissage, constituant les savoirs fondamentaux mais aussi les savoir-être et savoir-faire nécessaires à la conception de modèles paramétrique en conception architecturale,
- une capacité à apprendre, à mettre à jour ses connaissances dans la continuité, même après le temps du cours.

En cela une **pédagogie « de l'apprentissage »** semble adaptée à notre objet (Altet 2006, p.20). Dans une pédagogie de l'apprentissage l'objectif est d'amener l'étudiant à « apprendre à apprendre ». Cette pédagogie est centrée sur le rapport de l'apprenant au savoir et sur la médiation de l'enseignant.

Par ailleurs, la mise en place et l'utilisation lors du cours de **supports d'apprentissage en ligne** semble pouvoir être efficace pour rendre accessible aux apprenants des ressources permettant de poursuivre ou d'approfondir un apprentissage de la modélisation paramétrique.

### **8.1.2 Construction d'objectifs pédagogiques pour l'apprentissage de la modélisation paramétrique par des architectes**

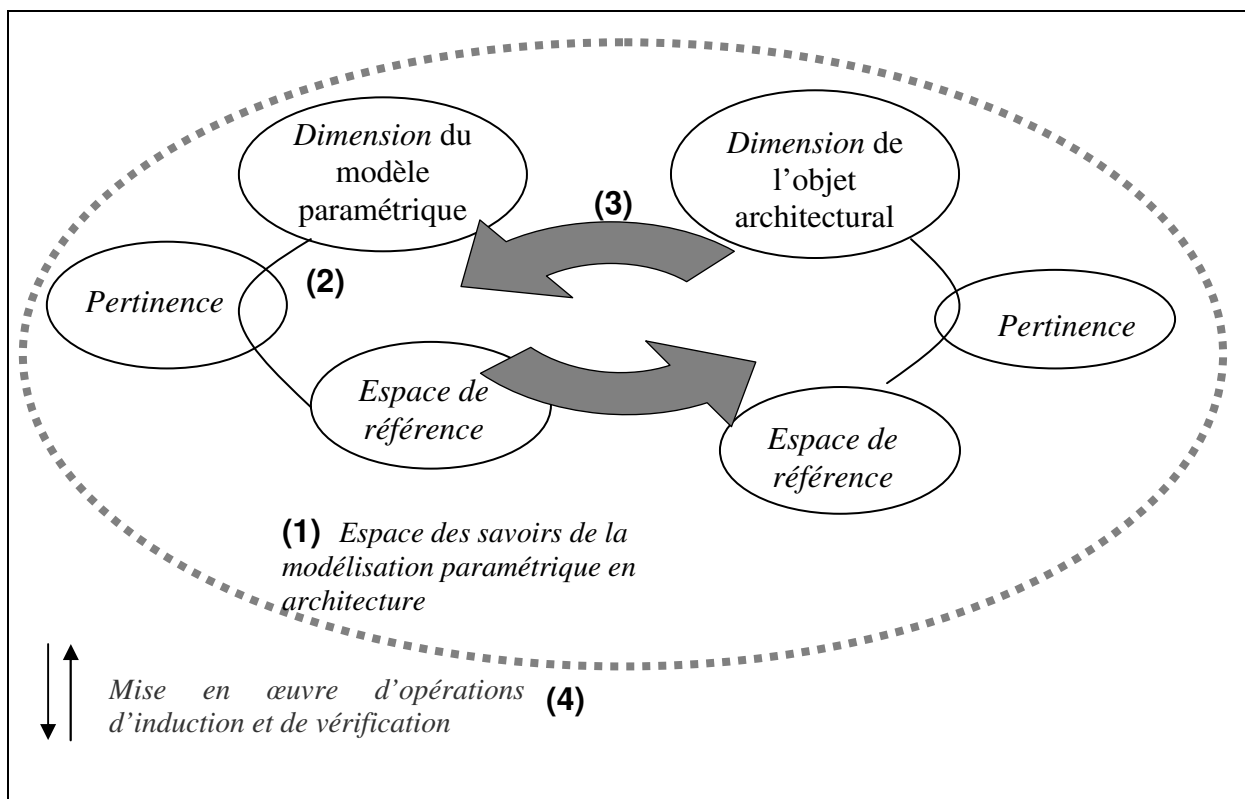
---

<sup>98</sup> Par exemple la mise à jour de septembre 2012 de Grasshopper étant la première en six mois, sa diffusion à fait l'objet d'excuses de la part de son développeur David Rutten.

<sup>99</sup> Par exemple les structures des données des composants ont été transformées lors de mises à jour majeurs de Grasshopper.

Nous avons précédemment situé le positionnement général de la pédagogie que nous proposons. Cette section présente à présent les objectifs pédagogiques que nous construisons grâce aux résultats de la recherche sur les activités cognitives et les ressources sollicitées à partir de la conception de modèles paramétriques par les architectes. Un objectif pédagogique exprime le résultat général visé par la formation prodiguée : les compétences, savoirs, savoir-faire et savoir-être que l'apprenant doit maîtriser à la fin de la formation (Meirieu 2004; Altet 2006). La définition d'objectifs pédagogiques permet par la suite d'accompagner la construction d'un enseignement ou d'une ressource didactique et, d'en évaluer l'efficacité par rapport aux buts initiaux.

Les compétences, savoirs, savoir-faire et savoir-être nécessaires à la modélisation paramétrique, constituent pour nous l'*espace des savoirs de la modélisation paramétrique en architecture*. Nous avons vu dans les chapitres 6 et 7 que les opérations de conception de modèles paramétriques (*découpage, dimensionnement et référenciation*, cf. Figure(2)) ainsi que les opérations tierces participant à l'entremêlement de la conception de modèles paramétriques et de la conception architecturale (*interprétation, mise en commun, appropriation de l'outil et traduction en géométrie paramétrique* cf. Figure(3) et (4)) nécessitaient la mise en œuvre d'un *espace des savoirs de la modélisation paramétrique* (cf. Figure(1)).



**Figure 113 : Schématisation de l'activité cognitive mise en œuvre lors de la conception de modèles paramétriques en architecture et des différents points à aborder lors d'un apprentissage de cette activité**

Dans la présente recherche, nous utilisons le terme « connaissance » pour désigner les contenus construits par la recherche scientifique et le terme « savoir » pour désigner les contenus que l'on souhaite voir acquérir par l'apprenant. Un « savoir » est alors entendu comme une connaissance acquise ou élaborée par un sujet grâce à l'étude ou l'expérience (Champy et al. 1994, p.891). Le savoir est

opérateur, il est lié à un pouvoir et à une mise en œuvre des contenus (Champy et al. 1994, p.891)<sup>100</sup>. Un savoir-faire est une capacité à modifier le réel selon une intention par des actes appropriés et en sollicitant des savoirs spécifiques (Champy et al. 1994, p.891)<sup>101</sup>. La notion de savoir-faire en tant que tel ne comporte pas de jugement : elle désigne des gestes maladroits autant que performants. Comme les savoirs, les savoir-faire ont été appris et acquis. L'observation de situations différentes, des essais-erreurs analysés et la confrontation avec de nouveaux problèmes sont des moyens d'acquérir de nouveaux savoir-faire (Champy et al. 1994, p.892). Un savoir-faire peut aussi être nommé « *connaissance procédurale* » (Le Ny in Champy et al. 1994, p.188), c'est un savoir « afin de ». Il s'oppose alors au savoir, aussi appelé « *connaissance déclarative* », qui est un savoir « à propos de » (Le Ny in Champy et al. 1994, p.188). Nous utilisons également les termes « savoir-être » et « compétence ». Un savoir-être est une compétence comportementale ou sociale, comme la curiosité, l'écoute ou la concentration. Une compétence est la capacité à mobiliser des ressources : savoirs, savoir-faire et savoir-être, en vue de résoudre une situation complexe (Altet 2006).

Dans la suite de ce texte, nous décrivons les savoirs (8.1.2.1), les savoir-faire (8.1.2.2), ainsi que les compétences et les savoir-être (8.1.2.3), qui constituent un *espace des savoirs de la modélisation paramétrique en conception architecturale*.

### 8.1.2.1 Savoirs « à propos de »

Comme nous l'avons vu et dit précédemment, de nombreux savoirs théoriques sont nécessaires pour la manipulation d'un modèle paramétrique et pour la conception de modèle paramétrique dans le cadre particulier de la conception architecturale. Les prochains paragraphes présentent les spécificités de ces savoirs.

- **Savoirs propres à la géométrie et aux mathématiques** : La géométrie et les mathématiques sont nécessaires à la mise en œuvre *d'opérations de traduction en géométrie paramétrique* (cf.6.4.2.2 et Figure 113 (3)). Ces savoirs concernent entre autres: la génération de géométries « élémentaires » (par exemple pour définir un plan, un point, une ligne, etc.), les transformations (booléennes, euclidiennes, etc.), les vecteurs, la trigonométrie, les équations de courbes et de surface, etc. (Pottmann et al. 2007; J. Burry & M. Burry 2012; Woodbury et al. 2007).
- **Savoirs liés à l'informatique en général et à la programmation en particulier** : Quelques savoirs généraux sur l'informatique semblent être nécessaires à la compréhension des enjeux des mécanismes paramétriques « basés sur la propagation ». Ces savoirs concernent des connaissances sur : la logique des niveaux de programmation et des plugins et quelques principes de base de la programmation : tests et récursivités en particulier. Pour un usage avancé de la modélisation paramétrique, l'approfondissement de ces savoirs sur la programmation est nécessaire. L'apprentissage d'un langage de scripting spécifique à un

---

<sup>100</sup> « d'abord apprendre, c'est-à-dire assurer l'acquisition par une activité mentale suivie, puis mettre en œuvre, interpréter, etc. » (Champy et al. 1994, p.891).

<sup>101</sup> Champy définit un savoir-faire comme une « disponibilité à modifier une partie du réel selon une intention, par les actes mentaux et gestuels appropriés » (Champy et al. 1994, p.891).

modeleur paramétrique peut permettre à un concepteur de développer ses propres fonctions par exemples.

- **Savoirs sur le numérique en général** : Concevoir un modèle paramétrique peut nécessiter d'un concepteur qu'il se situe par rapport aux enjeux plus généraux du numérique. Quelques repères peuvent être donnés à l'apprenant dans cette optique. Ces enjeux sont très larges : ils sont historiques (Picon 2010; Carpo 2012), philosophiques (Triclot 2011; Simondon 1958; Barthelemy 2008), sociaux (Casilli 2010) et peuvent également être liés à une certaine actualité. Par exemple, les questions de l'Open Data (publication en ligne de certaines données nationales) [<http://www.data.gouv.fr/>], du Big Data (gestion de très larges bases de données pour la recherche ou l'information en général), des nouvelles sociabilités numériques ou du partage en ligne sont actuellement autant d'éléments qui, sans être directement liés à une « géométrie numérique », peuvent enrichir les questionnements de l'apprenant. Par ailleurs, des repères sur le numérique en général peuvent aider à évaluer la portée de la modélisation paramétrique. Par exemple, lors de l'introduction de la modélisation paramétrique chez Gehry Partners, une forte résistance provenait d'un manque de connaissance des outils paramétriques et d'une croyance en la capacité de ces outils à « prendre le pouvoir » sur la conception (Smith 2005, p.42). Pour Smith et Glymph, introduire la modélisation paramétrique chez Gehry Partners a impliqué, dans une grande part, de changer la culture de l'agence (Smith 2005, p.42)<sup>102</sup>. L'enjeu est également de préparer l'étudiant au monde professionnel en lui permettant, entre autres, d'appréhender les nouveaux métiers liés à la modélisation paramétrique (De Kestelier in Kocaturk & Medjdoub 2011, p.241)<sup>103</sup>.
- **Savoirs sur les mécanismes « basés sur la propagation » de la modélisation paramétrique** : Pour concevoir un modèle paramétrique pertinent et organiser les dépendances entre des variables, un concepteur doit être capable d'appréhender le mécanisme d'un modeleur paramétrique pour comprendre sa logique de propagation (Woodbury 2010, p.25)<sup>104</sup> mais aussi les contraintes dues à son organisation en graphes acycliques et dirigés (cf. Chapitre 2). Ces savoirs participent directement à la mise en œuvre d'opérations de conception de modèles paramétriques (cf. Figure 113 (2)) et d'opérations d'*induction* et de *vérification* (cf. Figure 113 (4)).
- **Savoirs propres à un logiciel** : Comme nous l'avons déjà abordé dans le chapitre 2 lors de l'analyse des modeleurs paramétriques choisis (cf. 2.3, p. 47), la conception d'un modèle paramétrique nécessite des compétences spécifiques au modeleur utilisé. En effet, le concepteur doit connaître le fonctionnement de l'interface du modeleur, les principales fonctions proposées, les modes de traitements des données, *etc.* Par exemple, pour l'usage de Grasshopper, un

---

<sup>102</sup> “Trying to teach them these new concepts was hard. They wanted the technology to conform to their way of thinking. They kept telling me, —This is not how it's done in our industry. But I was there to change their methods. Working in three dimensions requires a rigorous knowledge of geometry and an engineering approach. As I intersected the complex geometries of Frank's designs, they were confused. They always asked me to show the views of the design in plan or section only. They just couldn't understand clearly what they were seeing otherwise. This confusion came from their training in school to think in two dimensions, transposing the flat views drawn on paper into a 3D image in their mind's eye.” (Smith 2005, p.42).

<sup>103</sup> « Part of our role is not only guiding the team on how make the tools but actually how to apply them in real context » (De Kestelier in Kocaturk & Medjdoub 2011, p.241).

<sup>104</sup> « Conceiving, arranging and editing dependencies is the key parametric task. » (Woodbury, 2010 p. 25).

concepteur doit connaître entre autres la programmation graphique, la constitution des arbres de données (cf. 2.3.1) et le mode de navigation au sein de l'espace de travail.

- **Savoirs liés à l'architecture et à la construction** : Tous les acteurs de la conception de modèle paramétrique en architecture avancent la nécessité, pour cette pratique, de connaissances en architecture : la plupart des experts de la modélisation paramétrique en conception architecturale sont avant tout des architectes. Il est difficile de circonscrire ces savoirs, propres à l'architecture et à la construction, mais il nous faut tout de même pointer leur nécessité.
- **savoirs liés à la conception architecturale** : Un savoir propre à la conception architecturale permettrait à l'apprenant de développer une réflexivité sur sa pratique de conception architecturale (Boudon et al. 2000). Cette réflexivité pourrait aider l'apprenant à développer des modèles paramétriques spécifiques au processus de conception architectural développé. Certains usages des apports de l'architecturologie pour l'enseignement permettent le développement de tels savoirs (Boudon et al. 2000; Lecourtois 2008).

Ces différents savoirs participent à l'espace des savoirs de la modélisation paramétrique en conception architecturale (cf. Figure 114).

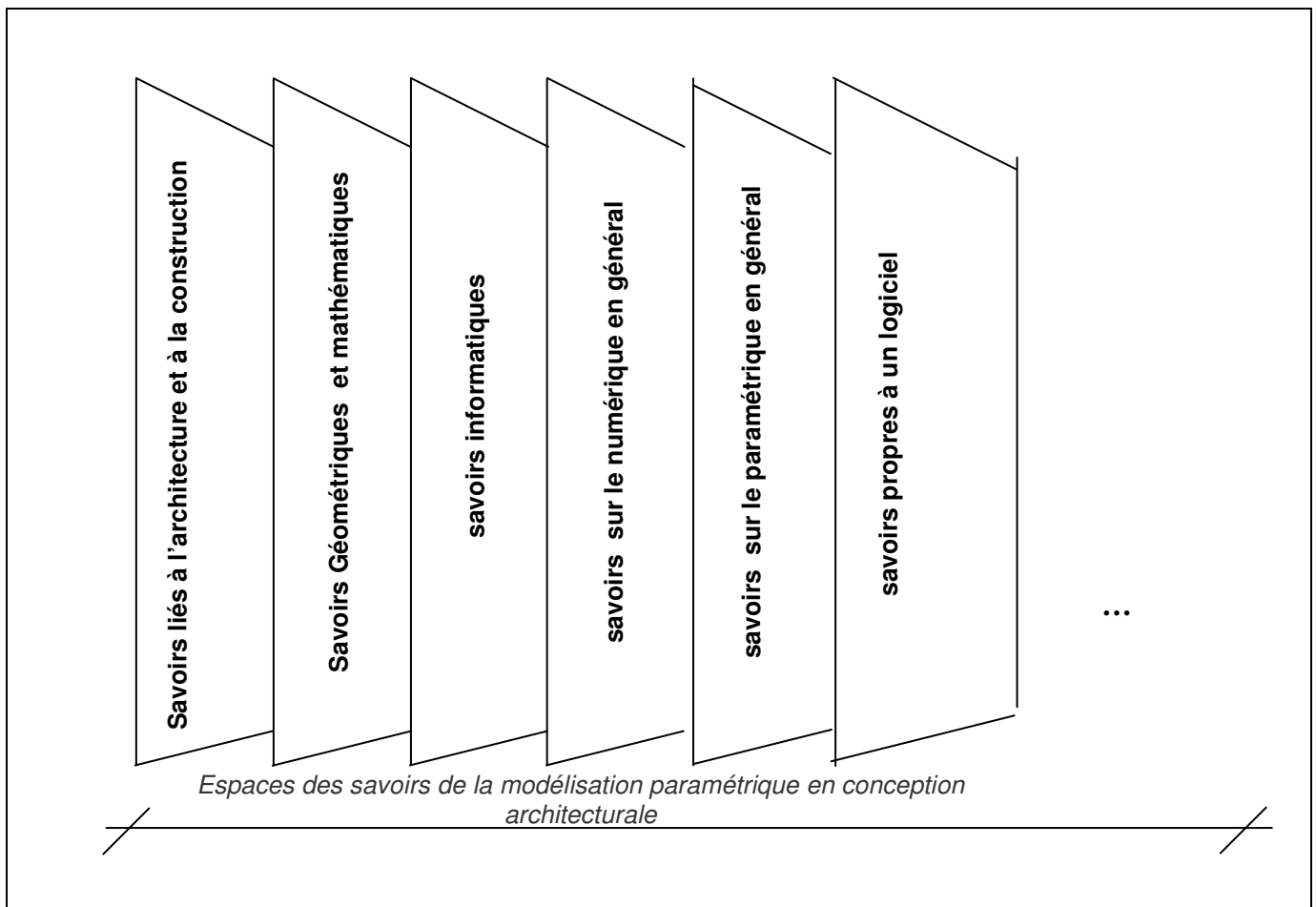


Figure 114 : Savoirs constituant l'espace des savoirs de la modélisation paramétrique en conception architecturale



### 8.1.2.2 Savoir-faire spécifiques à la modélisation paramétrique : les patterns

Nous venons de décrire les savoirs théoriques nécessaires à l'usage de la modélisation paramétrique en conception architecturale. Nous abordons maintenant les savoir-faire nécessaires à la conception de modèles paramétriques en conception architecturale. Pour définir ces savoir-faire, nous nous appuyons sur la méthode des patterns développée par Robert Woodbury pour la conception de modèles paramétriques en conception architecturale. Les patterns proposés par Robert Woodbury sont des stratégies de conception de modèles paramétriques spécifiques à la conception architecturale : ils donnent accès à des savoir-faire spécifiques à la modélisation paramétrique. L'état des savoirs relatifs à cette méthode se trouve dans la section 4.1 p. 106 et l'analyse de l'usage de cette méthode dans le chapitre 7, p. 235. Pour avoir accès à ces savoir-faire, nous avons vu que les patterns étaient une aide effective au travers de leurs applications sous la forme de samples<sup>105</sup>. Les patterns, au moyen des samples (cf. chapitre 7), donnent accès à des savoir-faire spécifiques à la modélisation paramétrique en conception architecturale, en pointant certains savoirs (cf. Figure 115).

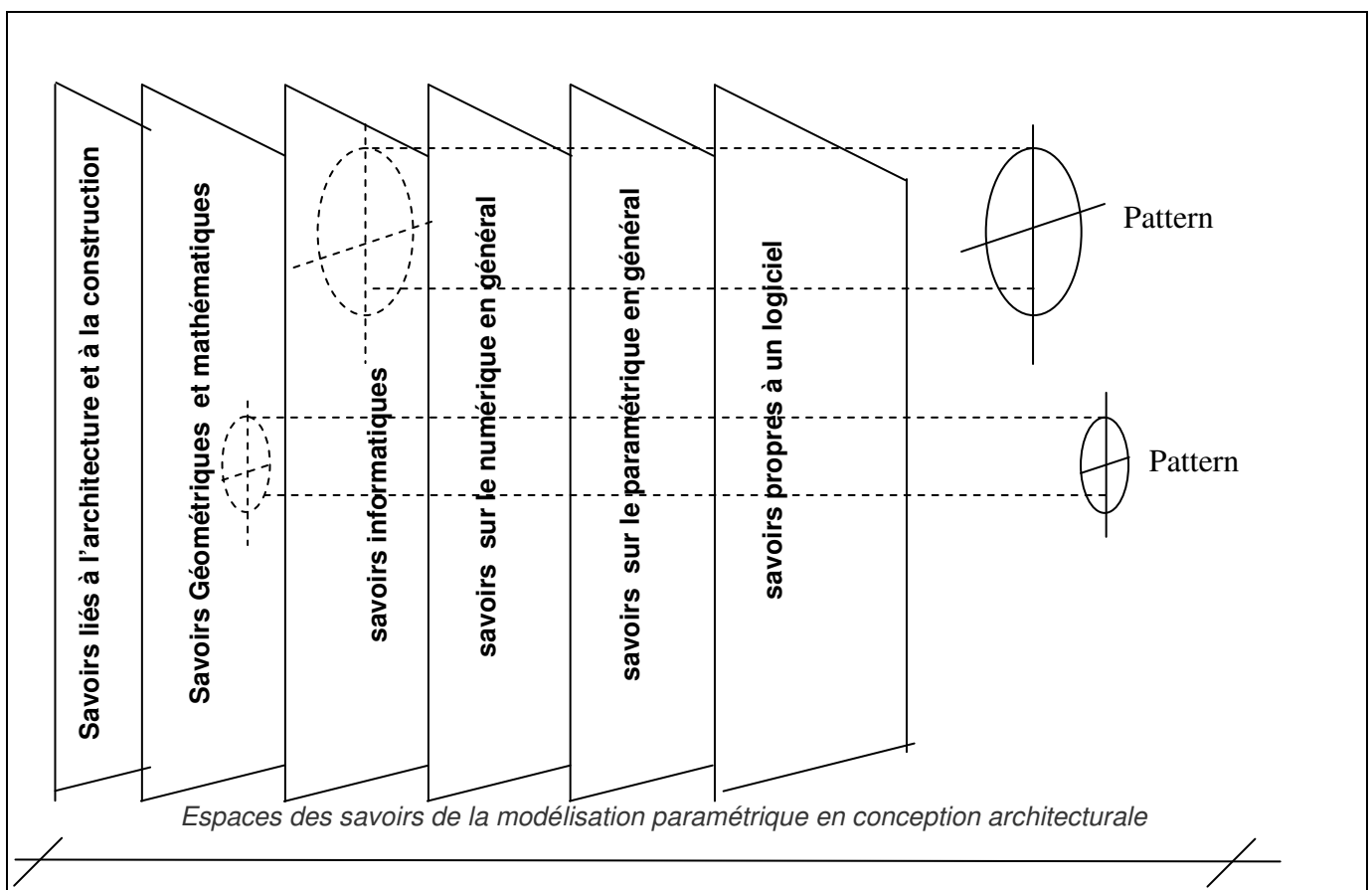


Figure 115 : Les patterns comme accès à des savoir-faire sollicitant des savoirs spécifiques

<sup>105</sup> Un sample est l'application d'un pattern à un problème de modélisation spécifique à la conception architecturale.

Aux savoir-faire auxquels donnent accès les patterns, Robert Woodbury associe la stratégie de modélisation qu'il nomme : « Divide and Conquer » (Woodbury 2010, p.188) (cf. 4.1.2 p.113). Cette stratégie consiste à diviser un projet de modèle paramétrique en sous-systèmes identifiables et compréhensibles, aux interactions simples, que l'on modélise ensuite en faisant appel à des patterns.

Par ailleurs, on peut pointer un savoir-faire lié à l'opération de *traduction en géométries paramétriques* (cf. 6.4.2.2 et Figure 113 (3)). Whitehead pointe la difficulté de ce savoir-faire lorsqu'il affirme lors d'un entretien : « *When someone join us, it takes about a year to build up that fluency of all the skills and techniques, and then it takes a further year before they can really take on a design brief and interpret it properly* » (Whitehead in Kocaturk & Medjdoub 2011, p.239). D'après lui, un concepteur de modèles paramétriques doit non seulement maîtriser un certain nombre de techniques et de savoirs et les mobiliser pertinemment, mais il doit aussi être capable de traduire des intentions architecturales, c'est-à-dire les *pertinences* et les *espaces de références* mobilisés dans la conception du projet architectural impliqué. L'*opération de traduction en géométries paramétriques* sollicite donc à la fois les divers savoirs exposés dans le paragraphe précédent (cf. 8.1.2.1), mais aussi un savoir-faire résultant d'expériences préalables de la traduction d'intentions architecturales en géométries paramétriques. Cette expérience, dans le cadre d'un enseignement, peut être introduite par l'expérimentation de telles traductions dans des projets personnels à l'apprenant ou dans des exercices ciblés.

### 8.1.2.3 Savoir-être et compétences nécessaires à l'usage de la modélisation paramétrique

Plusieurs acteurs de la modélisation paramétrique pointent l'importance « d'attitudes » spécifiques, ainsi Whitehead avance: « *I think the changes are more about attitudes than about technology and that comes with experience* » (Whitehead in Kocaturk & Medjdoub 2011, p.238). Selon nous, ces attitudes concernent des savoir-être et des compétences spécifiques à :

- **L'abstraction (logique et géométrique)** : Un concepteur de modèle paramétrique doit être capable de penser un modèle de façon abstraite, afin: -d'organiser un modèle en termes de parties logiques (cf. la stratégie « Divide and Conquer » proposée par Woodbury), -de concevoir un modèle avec un minimum d'entrées possibles en vue de faire un modèle générique, - d'interpréter une intention architecturale en termes de géométrie paramétrique. On voit donc que l'aptitude à l'abstraction est sollicitée lors de la mise en œuvre d'opérations de conception de modèles paramétriques (cf. Figure 113 (2)) mais aussi pour les opérations *d'induction* et de *vérification* (cf. Figure 113 (4)) et de *traduction en géométries paramétriques* (cf. Figure 113 (3)).
- **L'organisation et l'anticipation** : Une certaine adaptabilité d'un modèle paramétrique aux transformations (évolution ou réorientation du projet architectural) est déterminante pour la performance de celui-ci. En effet, un modèle paramétrique trop figé (ou auquel on ne peut pas faire subir les transformations voulues) ne sera ni intéressant ni rentable si sa construction a demandé beaucoup de temps et d'énergie. L'organisation d'un modèle et l'attention à sa lisibilité sont indispensables à la viabilité d'un modèle sur le long terme. Pour cela, le concepteur de modèles paramétriques doit développer une capacité d'organisation et

d'anticipation, participant à la mise en œuvre d'opérations de conception de modèles paramétriques (cf. Figure 113 (2)).

- **Participer, entretenir un réseau et travailler en équipe** : L'usage de la modélisation paramétrique pour être performant doit : -s'inscrire dans une logique d'auto-formation continue du concepteur (les logiciels évoluent en permanence) et, -s'inscrire dans une logique de réutilisation de modèles déjà conçus. Pour cela, l'inscription du concepteur dans une communauté d'utilisateurs peut beaucoup lui apporter. C'est dans cette logique qu'ont été créées certaines communautés comme Smart Geometry. Pour cela, le concepteur doit développer une capacité de participation et d'entretien d'un réseau et une compréhension de ce que celui-ci peut lui apporter. De même, une compréhension et une envie de partage (partage de compétences par l'aide sur un forum ou le partage de ressources par la diffusion de ces modèles) peuvent lui être profitables. Ces aptitudes permettent au concepteur d'entretenir et d'enrichir son espace des savoirs de la modélisation paramétrique (cf. Figure 113 (1)).
- **La curiosité et l'esprit d'initiative** : Le concepteur de modèles paramétriques doit pouvoir avoir un comportement d'exploration : il doit, entre autres, chercher à comprendre sans avoir peur des bugs, savoir où chercher l'information et avoir un intérêt pour la technologie.

### 8.1.2.4 Objectifs pédagogiques

Un *objectif pédagogique* exprime le résultat général visé par la formation prodiguée. Les objectifs pédagogiques expriment les compétences, savoirs, savoir-faire et savoir-être que l'apprenant doit maîtriser à la fin de la formation en termes d'activités. La définition des objectifs pédagogiques de l'enseignement que nous proposons permet par la suite d'accompagner sa construction ou celle des ressources didactiques proposées et d'en évaluer l'efficacité par rapport aux buts initiaux.

Nous proposons ici:

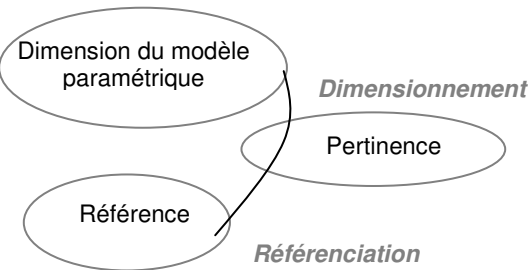
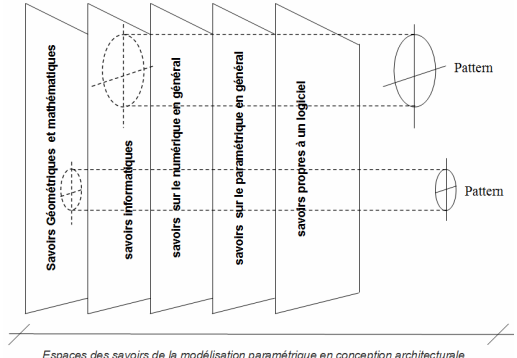
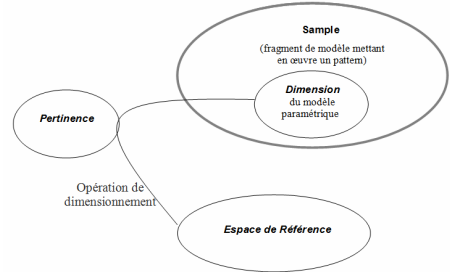
-des objectifs pédagogiques visant à accompagner la construction d'un *espace des savoirs de la modélisation paramétrique* (Tableau 36) ;

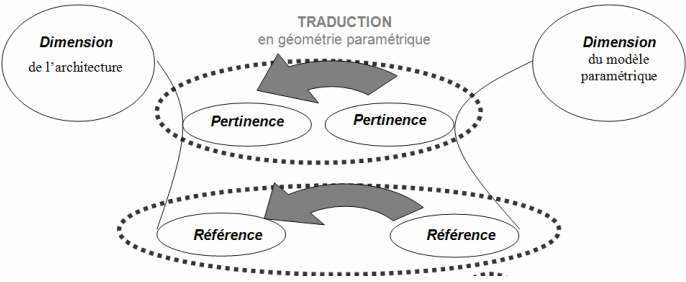
- des objectifs pédagogiques liés à la mise en œuvre des opérations cognitives spécifiques à la conception de modèles paramétriques en conception architecturale (Tableau 37).

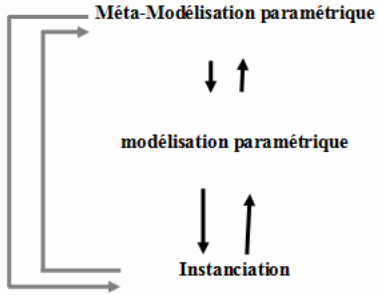
|   |   |
|---|---|
| Objectifs pédagogiques liés aux savoirs | -l'apprenant doit savoir où trouver une information s'il se trouve dans une situation inconnue ou devant une question technique (propre au modeleur ou propre à la géométrie ou l'informatique en général) qu'il ne maîtrise pas. |
|   | -l'apprenant doit maîtriser des bases de géométrie paramétrique (définir une surface développable, un plan, une courbe, etc.) et d'informatique (boucle, plugin)  |
|   | -l'apprenant doit posséder des repères sur les principaux enjeux liés au numérique faisant l'actualité de son futur cadre de travail  |

|  |   |
|--|---|
|  | -l'apprenant doit pouvoir choisir entre les principaux modeleurs paramétriques et leurs modes d'échanges avec les principaux moteurs d'évaluation, d'interaction et de fabrication, pour choisir le logiciel, ou l'agencement de logiciels, le plus approprié à son projet. |
| Objectifs pédagogiques liés aux savoir-faire                   | -l'apprenant doit pouvoir manipuler le modeleur paramétrique abordé lors de la formation pour concevoir un modèle paramétrique simple   |
|  | -l'apprenant doit pouvoir appliquer la stratégie « Divide and conquer » à la conception d'un modèle paramétrique simple   |
|  | -l'apprenant doit pouvoir organiser et rendre lisible un modèle paramétrique qu'il a construit  |
|  | -l'apprenant doit pouvoir lire un modèle paramétrique qu'il n'a pas lui-même développé et le modifier pour l'adapter aux objectifs de modélisation qui lui sont personnels  |
|  | -l'apprenant doit savoir faire du projet architectural et doit pouvoir développer une réflexivité sur sa pratique de conception   |
| Objectifs pédagogiques liés aux compétences et aux savoir-être | -l'apprenant doit pouvoir interpréter une intention architecturale simple en géométrie paramétrique   |
|  | -l'apprenant doit pouvoir organiser son modèle paramétrique en anticipant de futures évolutions de celui-ci   |
|  | -l'apprenant doit pouvoir organiser son travail, mais aussi écouter et interpréter un propos de façon à travailler en équipe  |
|  | -l'apprenant doit pouvoir participer à un réseau social d'utilisateurs de modeleurs paramétriques   |

**Tableau 36 : Objectifs pédagogiques liés à la construction d'un espace des savoirs de la modélisation paramétrique**

| Opérations cognitives impliquées   | Ressources cognitives sollicitées   | Objectifs pédagogiques impliqués   |
|--|---|--|
| <p><i>Découpage</i></p>  <p><i>Dimensionnement</i></p> <p><i>Référenciation</i></p> <ul style="list-style-type: none"> <li>- opérations de <i>découpage</i></li> <li>- opérations de <i>dimensionnement</i></li> <li>- opérations de <i>référenciation</i></li> </ul> | <p><i>Espaces des savoirs de la modélisation paramétrique en conception architecturale</i></p>  <p><i>Espace des savoirs de la modélisation paramétrique en conception architecturale</i></p>  <p>Aide aux opérations de découpage, dimensionnement et référencement par la méthode des patterns, et plus particulièrement par la médiation de samples</p> | <p>A la fin de la formation à la modélisation paramétrique en architecture, l'apprenant doit pouvoir :</p> <ul style="list-style-type: none"> <li>-décomposer une intention de modèle paramétrique en termes d'enchaînement logique et de dépendances géométriques,</li> <li>- dimensionner un modèle paramétrique en fonction de pertinences qui lui sont propres, en particulier en fonction de pertinences liées à la conception architecturale.</li> </ul> |

|  |   |   |
|--|---|---|
| <p>Mise en œuvre d'une <i>unité de conception</i> relative à une instance</p>  | <p><i>Espace des savoirs de la modélisation paramétrique en conception architecturale</i></p> <p>Et plus particulièrement : importances des savoirs relatifs à l'architecture et la construction</p>  | <p>-l'apprenant doit pouvoir tester la viabilité d'un modèle paramétrique simple en testant les extremums de son domaine de solutions</p> |
| <p>Mise en œuvre d'une <i>unité de conception</i> relative à un modèle paramétrique</p>  | <p><i>Espace des savoirs de la modélisation paramétrique en conception architecturale</i></p>   | <p>-l'apprenant doit pouvoir identifier les principales règles de génération d'un modèle à partir de cas particuliers simples</p>         |
| <p>Mise en œuvre d'une <i>unité de conception</i> relative à un méta-modèle paramétrique</p>   | <p><i>Espace des savoirs de la modélisation paramétrique en conception architecturale</i></p> <p>Et plus particulièrement : importances des savoirs liés à l'informatique en général, aux mécanismes « basés sur la propagation » ainsi qu'au modèleur utilisé</p>                |   |
|  <p><i>-Opération de traduction en géométrie paramétrique</i></p> | <p><i>Espace des savoirs de la modélisation paramétrique en conception architecturale</i></p> <p>Et plus particulièrement : importances des savoirs en géométries et mathématiques, en architecture et construction ainsi que des savoirs liés à la conception architecturale</p> | <p>-l'apprenant doit pouvoir interpréter une intention architecturale simple en géométrie paramétrique</p>                                |

|  |   |  |
|--|---|--|
| <ul style="list-style-type: none"> <li>- Opérations d'interprétation pour la collaboration</li> <li>- Opérations de mise en commun</li> <li>- Opérations d'appropriation de l'outil</li> </ul>                             | <p><i>Espace des savoirs de la modélisation paramétrique en conception architecturale</i></p>   | <ul style="list-style-type: none"> <li>-l'apprenant doit pouvoir interpréter les propos d'une tierce personne ou trouver des informations spécifiques à un problème nouveau, pour mettre ces informations en œuvre dans la conception de nouveaux modèles</li> <li>-l'apprenant doit être capable de choisir l'outil ou la fonction adaptée à la tâche de modélisation qu'il s'est fixé</li> </ul> |
| <div style="text-align: center;">  <p><i>opérations logiques d'induction</i></p> <p><i>Opérations logiques de vérification</i></p> </div> | <p><i>Espace des savoirs de la modélisation paramétrique en conception architecturale</i></p> <p>Et plus particulièrement : importances des savoirs sur les mécanismes paramétriques « basés sur la propagation »</p> | <ul style="list-style-type: none"> <li>-l'apprenant doit pouvoir tester la viabilité d'un modèle paramétrique simple en testant les extremums de son domaine de solutions</li> <li>-l'apprenant doit pouvoir identifier les principales règles de génération d'un modèle à partir de cas particuliers simples</li> </ul>   |

**Tableau 37 : Objectifs pédagogiques liés aux opérations spécifiques à la conception de modèles paramétriques en conception architecturale**

A partir de la définition de ces objectifs, nous proposons dans la suite de ce chapitre des ressources didactiques visant à assister l'accès à un *espace des savoirs de la modélisation paramétrique en conception architecturale* pour l'apprenant.

## 8.2 Supports didactiques pour l'apprentissage de la modélisation paramétrique en conception architecturale

La didactique est un processus pratique dont la visée est de transmettre des connaissances (scientifiques) en les transformant en savoirs (savoirs, savoir-faire et savoir-être) (Altet 2006). La didactique désigne aussi le travail de préparation des connaissances en vue de leur communication et, c'est sur cet aspect là de la didactique que nous nous attarderons.

Deux supports didactiques sont proposés pour accompagner l'enseignement de l'usage de la modélisation paramétrique aux architectes. Ces supports s'appuient sur : -la méthode des patterns proposée par Robert Woodbury et dont l'analyse nous a permis de repérer les limites et les potentialités ainsi que sur, -la définition que nous avons proposée précédemment d'un *espace des savoirs de la modélisation paramétrique*. Ces supports sont publiés en ligne : ils visent à accompagner une possible formation continue de l'apprenant (cf. Figure 116), tel que défini en 8.1.1 p. 261.

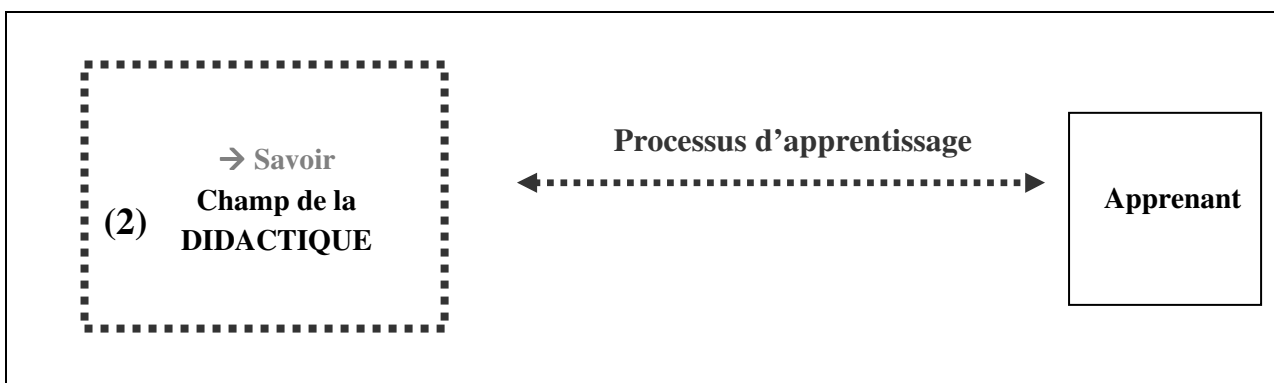


Figure 116: Mise en place d'une didactique de la modélisation paramétrique rendant possible la formation continue de l'apprenant *via* des ressources en ligne

Si ces supports doivent pouvoir permettre à l'apprenant d'approfondir sa formation par lui-même, ils doivent également supporter la pédagogie que nous proposons en participant à la construction de la situation enseignant-apprenant (cf. Figure 117).



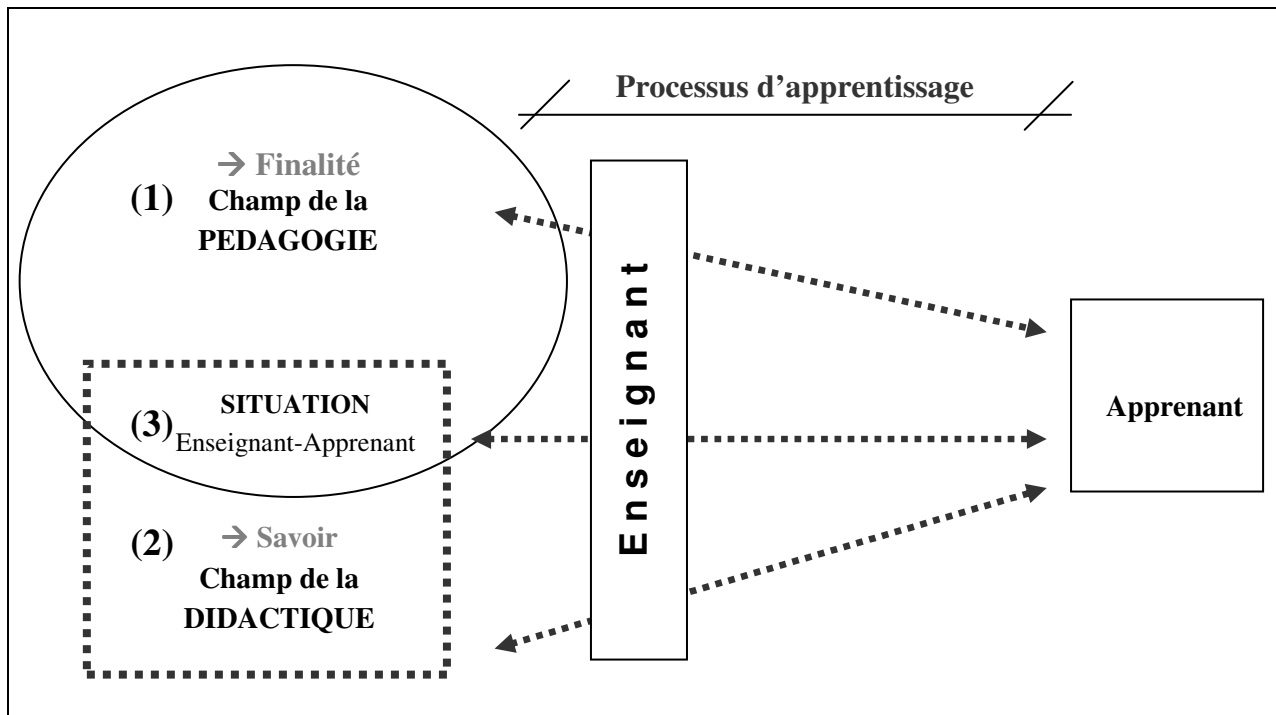


Figure 117: Proposition d'une pédagogie de la modélisation paramétrique pour l'architecte, centrée sur l'apprentissage

Ces supports doivent donc :

- être accessibles à distance ainsi qu'être disponibles et utilisables lors de cours en présence,
- supporter l'apprentissage grâce à la méthode des patterns et rendre accessible de façon plus large un *espace de la modélisation paramétrique en conception architecturale*.

Les supports didactiques que nous proposons pour cela sont :

- [parametric-ressources] : une plateforme de publication en ligne de samples liés aux patterns de Woodbury (cf. 8.2.1) ; et
- [dnarchi] : une plateforme de publication en ligne d'articles portant sur des questions relatives à « l'architecture numérique » en général (cf. 8.2.2).

### 8.2.1 [parametric-ressources] : publication en ligne de samples

[parametric-ressources] vise à rendre public des samples utiles à l'apprentissage de Grasshopper (cf. 8.2.1.1). La publication de ces samples, leurs lisibilités ainsi que les informations qui leurs sont associées, font l'objet d'un travail particulier décrit dans les paragraphes suivants.

### 8.2.1.1 Construction de samples pour l'usage de Grasshopper

L'analyse de la méthode des « Patterns » de Woodbury sous l'angle de la conception architecturale montre que si cette méthode est prometteuse, elle est rendu effective par son application dans des samples. A un pattern peut correspondre une multiplicité de samples (cf. Figure 118).

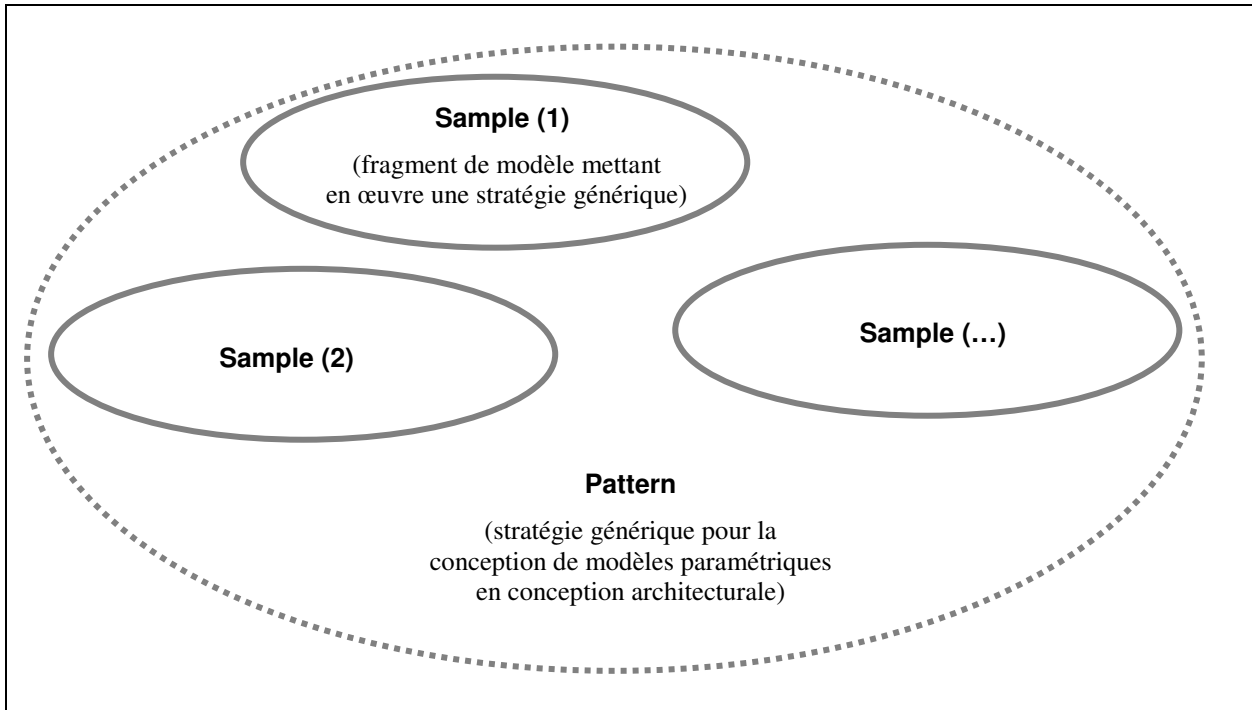


Figure 118 : Relation entre patterns et samples

Dans le cadre de l'expérimentation, nous avons construit une vingtaine de samples dont nous avons soumis l'usage à des étudiants. Nous avons vu dans le chapitre 7 que les samples proposés permettaient à la fois d'assister :

- la construction d'un *espace des savoirs de la modélisation paramétrique* (cf. Figure 119);
- et la mise en œuvre d'*opérations de conception de modèles paramétriques* (cf. Figure 120).

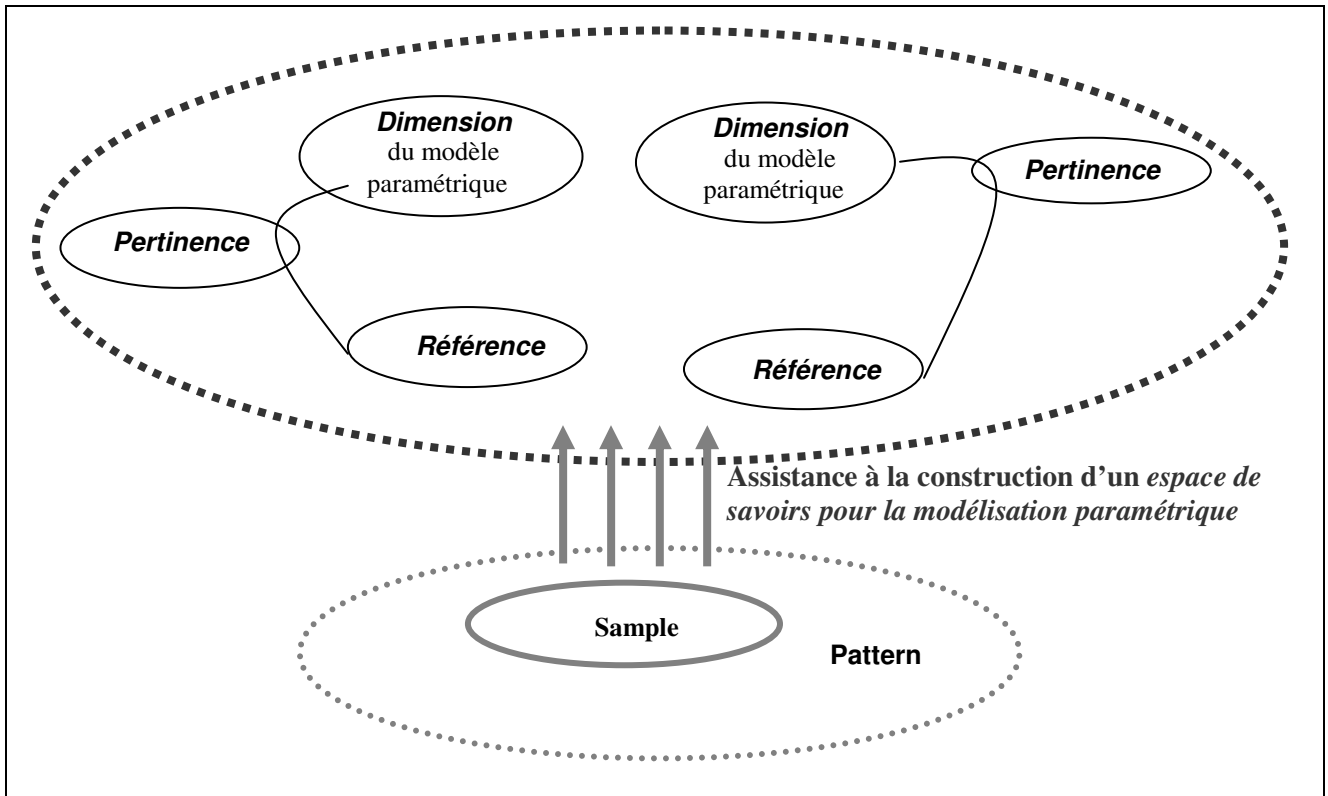


Figure 119 : Usage d'un sample comme assistance à la conception de modèle paramétrique en conception architecturale

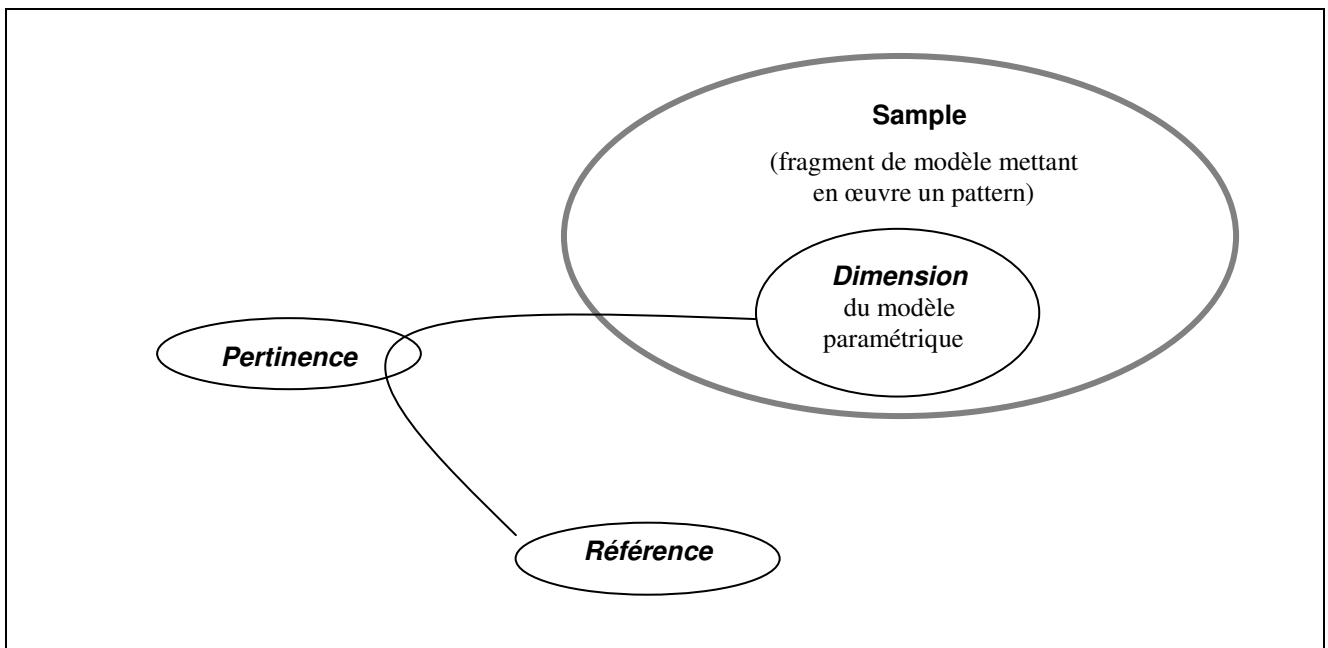


Figure 120 : Utilisation d'un sample dans une opération de dimensionnement directement en tant que fragment de modèle sur lequel porte la mesure

Les Samples construits lors des expérimentations répondent à deux critères (cf. 7.1.1 p. 236) :

- ce sont des applications de patterns de Woodbury ;

- ils répondent à des problèmes techniques propres à la modélisation paramétrique et récurrents en conception architecturale.

Le tableau qui suit rend compte des samples développés avec les patterns et des questions techniques qui leur sont relatives.

| <b>Samples</b>                 | <b>Patterns associés</b>               | <b>Questions techniques abordées</b>                        |
|--------------------------------|--|---|
| Extrait_item_liste4            | Place Holder                           | Motif, set, arbre de données                                |
| Mapping_list                   | Selector                               | Mapping, Arbre de données                                   |
| Extrait_valeur-max-min         | Selector                               | Sort List, Arbre de données                                 |
| creer-trame-pts_UV             | Points collection                      | Motif, trame, Point, sous-surface                           |
| Creer_pt_UV                    | Reactor                                | UV, Eval Pt on Srf, Reactor, Controller                     |
| creer_liste-sous-srf_UV        | Place Holder                           | Motif, sous-surface, Composant, Place Holder                |
| creer_pts_aleatoires           | Controller                             | Random, Point   |
| Creer_crv_math_spirale         | Points collection                      | Spirale, fonction, Courbes, Point, équation paramétrique    |
| Creer_crv_math_spirale 3d      | Point collection                       | Spirale, fonction, Courbes, Point, équation paramétrique    |
| Creer_crv_math-balle-tennis    | Point collection                       | fonction, Courbes, Point, équation paramétrique             |
| Creer_crv_sections-srf         | Reporter                               | Section, Intersection, Courbe                               |
| Creer_crv_RandomVoronoi        | Point collection                       | Voronoi, courbe   |
| creer_maillage_random-delaunay | Point Collection                       | Delaunay, Random, Mesh, Point Collection                    |
| creer_srf_repulseur            | Reactor<br>Point Collection<br>mapping | Reactor, srf, mapping, attracteur                           |
| Creer_srf_sections             | Reporter                               | Evaluer, Surfaces constructibles, Intersections             |
| Creer_srf_mathShell            | Point Collection                       | Coquillage, fonction, surface, Point, équation paramétrique |
| creer_motif-maillage           | Place Holder                           | Motif, composant  |
| Creer_motif-reactif_attracteur | Reactor<br>Place Holder<br>Mapping     | Mapping, Motif, attracteur                                  |
| Creer_motif-morphing           | Reactor<br>Place Holder                | Morphing, Motif   |
| Evaluer_ecart-type             | Reporter                               | Evaluer, fonction, écart type                               |
| Optimiser_vers-cible ?         | Goal Seeker                            | Goal Seeker, evaluer, optimiser, galapagos                  |
| Evaluer_heliodon               | Projection                             | Evaluer, ensoleillement, vector, Projection                 |
| evaluer_verticale-max          | Selector                               | Evaluer, angles entre vecteurs                              |

## Tableau 38 : Liste des samples développés dans le cadre de la thèse

La description complète des samples et des questions techniques traitées est insérée dans l'annexe 3 p. 417. Dans cette section, nous nous intéressons plus particulièrement à la formalisation des samples et à leur publication dans le cadre d'un enseignement de la modélisation paramétrique.

La plateforme [parametric-ressources] a été construite en janvier 2013 suite aux analyses des usages des samples et de leurs formalisations successives pour les Expe 1, 2 et 3. L'usage de [parametric-ressources] a été testé lors de l'Expe 4 ainsi que lors de l'enseignement de projet SP31 à l'ENSA de Versailles (sous la responsabilité de Claire Petetin).

La plateforme [parametric-ressources] vise :

- à rendre les samples visibles, lisibles et accessible à tous, même hors des cours en présence ;
- à inscrire les samples dans un *espace des savoirs de la modélisation paramétrique* plus large, par la mise en place de liens (références, liens hypertextes, renvois) vers des articles, des sources et des ressources diverses.

### 8.2.1.2 Rendre les samples visibles et lisibles

Si les samples sont des fragments de modèles, plusieurs informations y sont liées et enrichissent leurs usages, comme les informations sur les patterns qu'ils appliquent ou les questions techniques qu'ils résolvent. Comment présenter au mieux des samples pour rendre leurs usages possibles comme assistance à la mise en œuvre d'opérations de modélisation paramétrique et comme assistance à la construction d'un espace des savoirs de la modélisation paramétrique ? est l'objet de cette section.

#### - Didactique des samples : visualiser le modèle lui-même

Lors des Expe 1 à 3, on a pu observer que des samples étaient parfois reproduits (c'est-à-dire modélisés de nouveau) plutôt que copiés-collés à partir du fichier fourni. Dès l'Expé 2, des images des agencements des modèles étaient donc associées au catalogue des samples. Ces images sont utilisées par les étudiants dans leur usage des samples. A cette visualisation directe du modèle est associée une organisation spécifique du modèle visant à le rendre le plus lisible possible.

Les modèles eux-mêmes ont été structurés de manière à être lisibles et à fournir un maximum d'informations. Commentaires, organisations par groupe, titres et explicitations des entrées et des sorties ont été mis en avant dans chacun des modèles diffusés. Par exemple, dans le sample *Extrait\_1pt\_liste4* (cf. Figure 121), les entrées sont rendues lisibles par leur explicitation au travers de l'utilisation de composants spécifiques (cf. Figure 121-A). Des commentaires et une organisation des étapes du modèle par « groupes » (cf. Figure 121-B) guident la lecture du modèle. Des composants explicitent également les différentes sorties du modèle pouvant être intéressantes (cf. Figure 121-C).

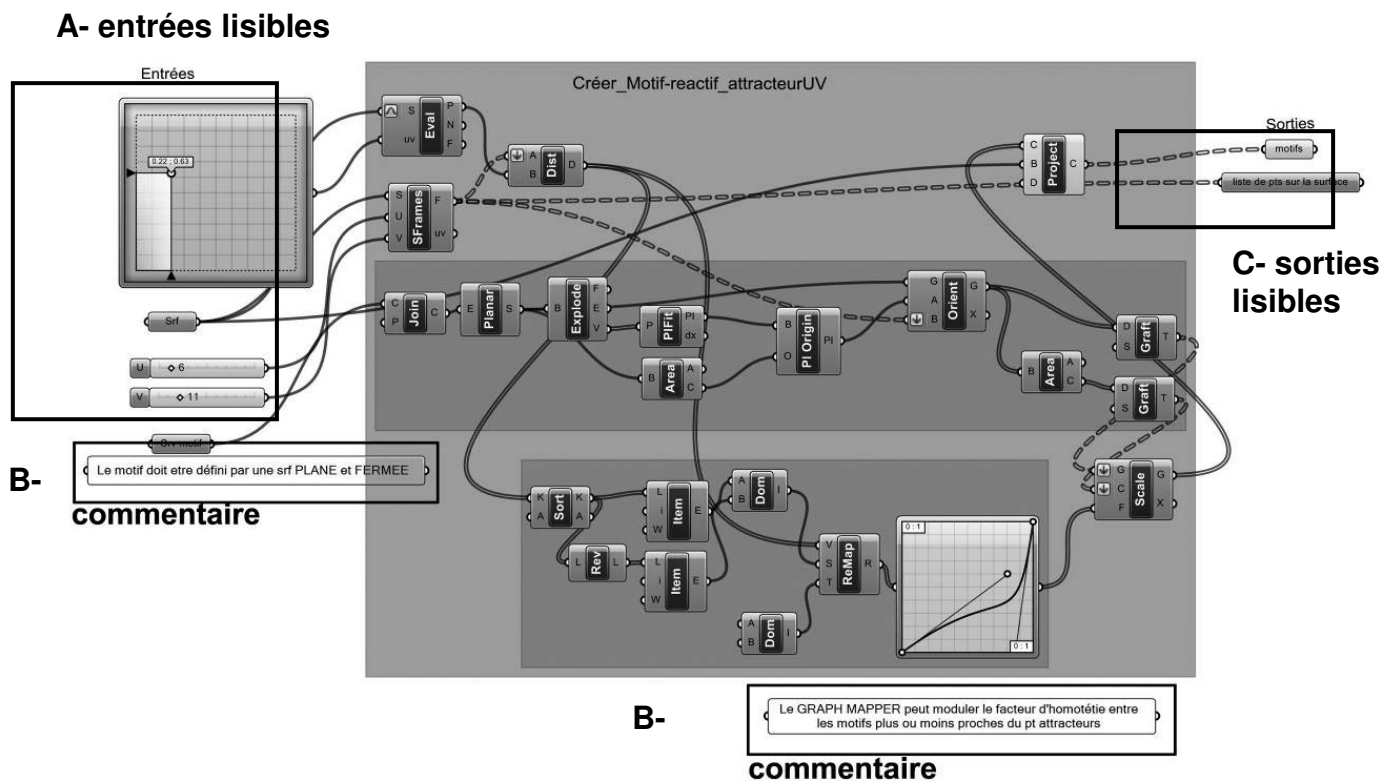


Figure 121 : Mise en forme d'un sample

**- Didactique des samples : informations associées**

Lors des Expé 1 et 2 les textes de présentation des samples ont été très peu lus. Ils ont alors été revus pour les rendre explicites et synthétiques (cf. Figure 122-A). Par ailleurs, les expérimentations menées ont montré l'importance du recours aux supports de cours et autres ressources en lien avec les samples. Les informations relatives à un sample (notions de géométrie ou de programmation, tutoriel, etc.) sont relayées par la plateforme sous la forme de courtes annotations ou de liens hypertextes vers des contenus précis (cf. Figure 122-B). Ainsi les samples sont contextualisés par rapport à des savoirs plus larges. D'autre part, des exemples d'utilisations des samples et des renvois vers d'autres samples qui lui sont relatifs sont également intégrés à la description de chaque sample (cf. Figure 122-C).

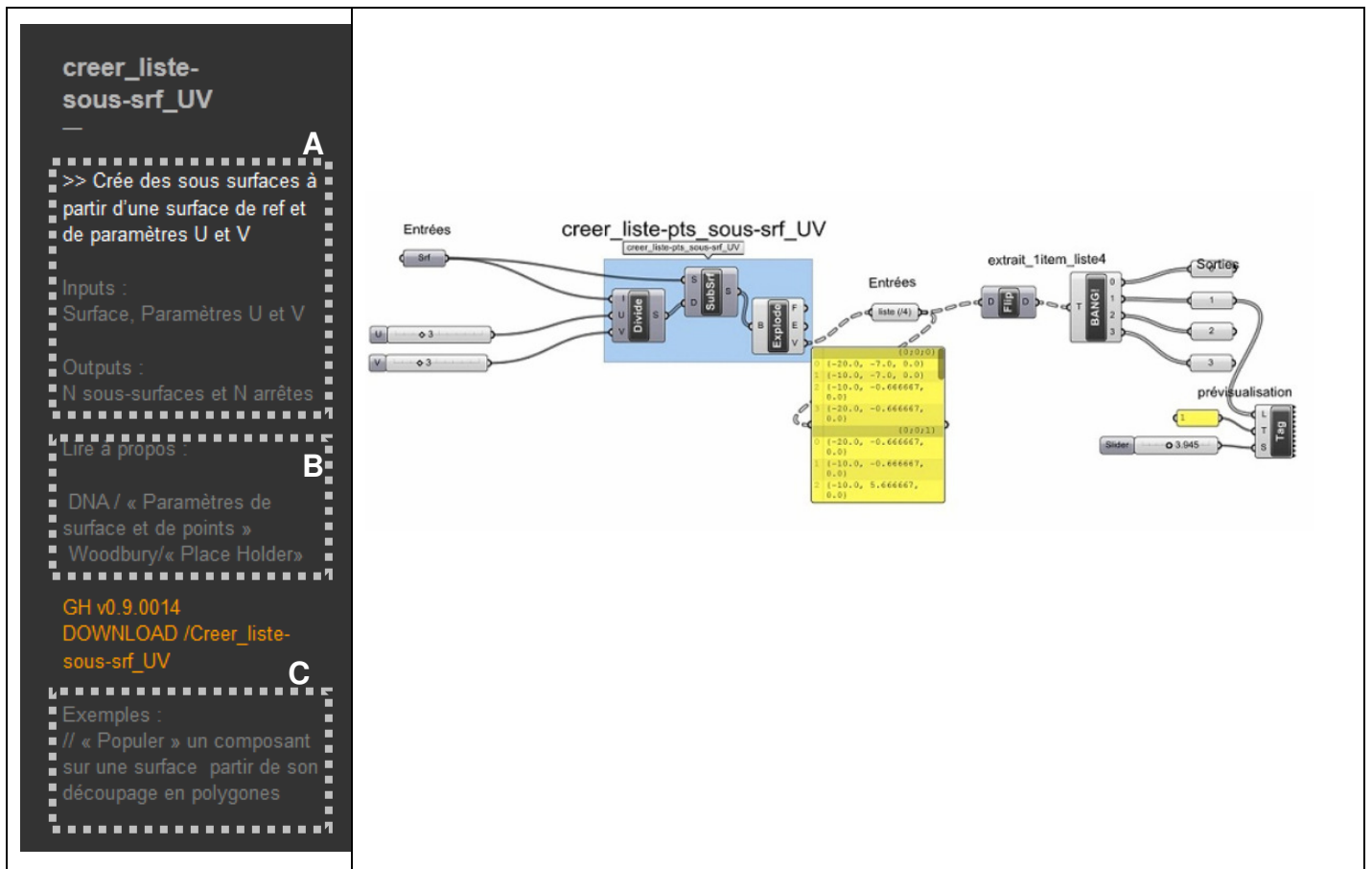
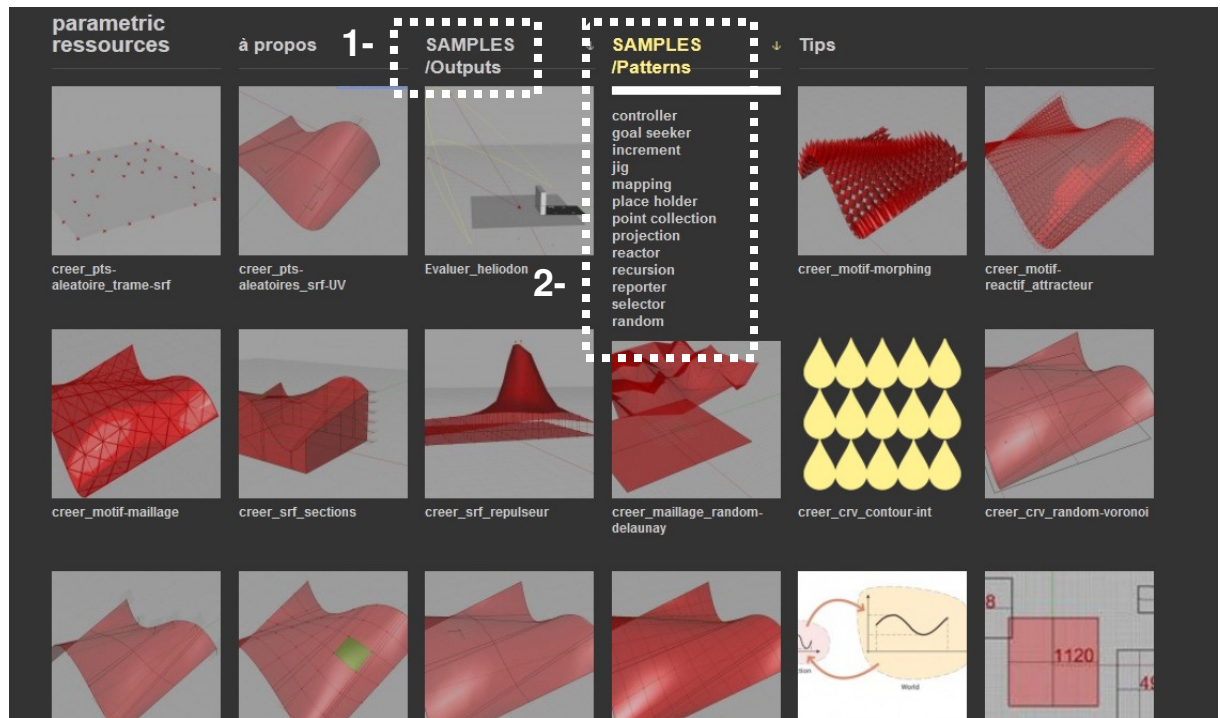


Figure 122 : Description d'un sample sur [parametric-ressources]

**- Chercher un sample dans le catalogue**

Par défaut, les samples sont présentés sur la page d'accueil du site dans l'ordre chronologique de leur publication (cf. Figure 123). L'objectif étant d'implémenter continuellement la base de données de samples, cette organisation permet aux visiteurs réguliers de retrouver les samples qu'ils connaissent et cela met en avant les nouveaux samples publiés.



**Figure 123 : Présentation a priori des samples par ordre chronologique et catégorisations « par outputs » et « par patterns »**

En outre, les samples sont classés par catégorie. Les catégories « par Outputs » (c'est-à-dire par sorties) définies lors des versions 1 et 2 du catalogue ont été conservées. Ces catégories renvoient aux géométries générées par le sample. Ces catégories sont : tree (pour arbres de données), points, courbes, surfaces-meshes, evaluation (cf. Figure 123-1). [parametric-ressources] intègre également un classement des samples par pattern (cf. Figure 123-2). Les Samples peuvent ainsi être recherchés en fonction de la stratégie de modélisation qu'ils permettent (sélectionner un item en fonction de ses propriétés pour le pattern selector, faire réagir un élément du modèle à sa proximité avec un autre élément pour le pattern reactor, etc.).

### **- Inclusion d'informations complémentaires**

Des informations techniques ou pratiques complémentaires aux samples sont intégrées dans les catégories « Tips » et « Tutorials » de [parametric-ressources]. Dans la catégorie « Tips », sont inclus des articles d'aides diverses comme l'article « liens utiles », renvoyant vers d'autres plateformes de ressources sur la modélisation paramétrique (cf. 4.2.1) ou encore l'article « hardware requirement » donnant des informations pratiques sur le matériel nécessaire à l'utilisation de Grasshopper.



## 8.2.2 Support didactiques : [dnarchi] Plateforme de publication de savoirs

Si [parametric-ressources] diffuse des aides techniques spécifiques (les samples et les « tips ») liées aux stratégies de modélisation des patterns, de nombreux savoirs autres participent à *l'espace des savoirs de la modélisation paramétrique*.

Nous avons esquissé les contours de cet espace des savoirs dans la section 8.1.2 (savoirs relatifs à la géométrie, à l'informatique, culture du numérique, etc.). Le contenu même de cet espace des savoirs résulte, pour une part, de connaissances scientifiques diffusées et reconnues (comme les connaissances fondamentales en géométrie ou en informatique) et, d'autre part, de connaissances scientifiques en construction peu diffusées en dehors des milieux spécialisés (comme l'état des pratiques actuelles ou encore certaines avancées en géométrie appliquée).

Ces savoirs, issus de la recherche, existent et sont publiés sur des modes spécifiques à la recherche (communication scientifique, article de revue, essais, etc.). Ces modes de publication (les sources comme l'écriture) ne sont pas accessibles pour des apprenants n'ayant pas déjà de solides connaissances dans les champs sollicités. Comment rendre accessibles aux apprenants, étudiants en architecture comme architectes praticiens, les résultats de la recherche en architecture numérique ? Nous avons proposé et mis en place une plateforme de publication à visée scientifique, pour la diffusion et le partage de connaissances relatives à la conception de l'architecture avec des outils numériques. Cette plateforme est [dnarchi] (cf. Figure 124). Aujourd'hui la publication en ligne est un enjeu de la diffusion de la recherche. Ce phénomène est interrogé par le champ des Digital Humanities.



Figure 124: Page d'accueil de [dnarchi]

La plateforme [dnarchi] permet de rendre accessible des articles donnant accès à:

-des résultats issus de recherche en cours, cela permet aux étudiants de percevoir la diversité des recherches effectuées mais aussi de percevoir certains points précis très spécialisés, comme dans l'article de Marion Roussel « *De l' « Alien » à l' « Allo- »*. *Devenir-alien, allogenèse et vie artificielle dans la transarchitecture de Marcos Novak* » (Roussel 2012).

- des présentations de structures existantes (surtout des centres de recherche ou des formations), comme dans l'article « *Le MAP, un laboratoire pluridisciplinaire et multi-sites* » de Halin et De Luca (2012) ou encore dans l'article « *Quelques remarques sur les enjeux pour toute école d'architecture de la « (non)représentation » à l'ère du numérique* » de Morel (2012) présentant un enseignement développé à l'ENSA de Paris Malaquais.

- des présentations de pratiques de projets mettant en œuvre des outils numériques, comme dans le cas de l'article « *Le modèle comme acte créatif et innovation combinatoire, Le projet Onion Pinch* » de Catarina Tiazzoldi (2012) ou encore l'article « *Network Thinking et apprentissage social: vers une architecture en réseau* » de Francesco Cingolani (2012).

-des éléments de connaissances « de bases » produites dans des visées pédagogiques, comme dans le cas de l'article « *Grasshopper et la programmation sur Rhinocéros 4: une introduction* », par de Boissieu et Guéna (2012), qui vise à situer le modelleur Grasshopper par rapport aux mécanismes de modélisation paramétrique et de la programmation visuelle.

Cette plateforme vise à être enrichie, non seulement en vue de participer aux débats scientifiques sur la connaissance « conception architecturale numérique », mais aussi en vue de nourrir la construction d'un *espace des savoirs de la modélisation paramétrique* (cf. 8.1.2 p. 262). Pour diffuser ces savoirs, des enjeux de fond (quoi publier ?) et de forme (comment le publier ?) ont été situés pour constituer la ligne éditoriale de la plateforme.

#### - Contenu de [dnarchi]

Les publications de [dnarchi] visent une qualité scientifique. Tous les articles sont relus par deux relecteurs experts du sujet. Un comité de rédaction oriente les thématiques de publication du site, par rapport auxquelles il sollicite des auteurs pertinents et accepte ou non des propositions d'articles. Les thématiques du site sont exprimées par les « catégories » des articles. Le Tableau 39 donne un aperçu du contenu de [dnarchi].

| <i>Nom affiché</i>          | <i>Contenu</i>  | <i>Articles (publiés ou en cours)</i>  |
|-----------------------------|---|--|
| <b>1/ Outils numériques</b> | # Articles interrogeant les usages et les rôles de technologies numériques pour « assister » la conception architecturale | « CAO : assistances et outils ? »<br>« GH et niveaux de programmation sur Rhino 4 »<br>« Structures de données en programmation »<br>« Interopérabilités avec GH »<br>« Les surfaces développables » ; « La géométrie dynamique » ; « La statique graphique »<br>« Collaborations : quels outils pour quels partages ? » |
| <b>2/ Culture numérique</b> | # Interroge largement la relation de l'architecte au numérique<br># point de vue historique,                              | « Architecture et Simondon »<br>« Coopération, Collaboration ou participation ? »<br>« La culture du libre »<br>« B comme Blob »   |

|                                 |   |   |
|---------------------------------|---|---|
|                                 | philosophique, etc.   |   |
| <b>3/ Pratiques numériques</b>  | # étude de cas de pratiques professionnelles<br># Articles interrogeant la conception à plusieurs, le rôle des communautés et des réseaux dans le numérique en archi  | « Quelles hybridités des pratiques »<br>« Nouveaux métiers du numérique en agence d'architecture »<br>« philosophie de participation contemporaine - network thinking - appliqué à dreamhamar »<br>« Taichi S: Description d'une pratique »   |
| <b>4/ Pédagogie et méthodes</b> | # qu'est ce que faire de la recherche en architecture<br># quid du numérique dans la recherche (digital humanities)<br># les différentes formations académiques disponibles sur l'archi numérique<br># l'enseignement et le numérique : transformation des pratiques pédagogiques | « La numérique et la pratique de la recherche »<br>« Blogging scientifique »<br>« Quelques repères sur l'historique de la recherche architecturale en France »<br>« nouveaux formats d'apprentissage: parametric design and networked learning »<br>« Communauté, réseau et ressources dans l'apprentissage de la modélisation paramétrique »<br>« Les Grands Ateliers de l'Isle d'Abeau : Quelle expérimentation des nouvelles technologies ? »<br>« Le Studio P9 à Malaquais, intégrer la FAO dans la conception »<br>« SP 31 à Versailles : réflexions « architecturables » »<br>« Nul n'entre ici s'il n'est géomètre » |
| <b>5/ Actualités</b>            | Critique d'expo, retours de colloques, sortie de nouveaux livres sur le numérique, diffusion d'appel à communication, etc.  | « CR expo le laboratoire »<br>Etc.  |

**Tableau 39 : Contenus publiés sur [dnarchi] par "catégorie" d'article**

#### **- Lisibilité du contenu de [dnarchi]**

La plateforme est accessible de façon rapide (*via* internet) et libre (les contenus sont directement accessibles sans nécessiter d'inscription ou de paiement par exemple). De plus, l'organisation visuelle de la plateforme (couleurs, page d'accueil, illustration des articles) et de l'accès aux articles, par métadonnées liées aux catégories et aux mots clefs, visent à rendre visibles les différents sujets traités par les articles.

Les articles ne dépassent pas 12 000 signes, doivent avoir un ton plus journalistique qu'académique et comprendre au minimum une illustration.

#### **-Futur et gouvernance de [dnarchi]**

L'initiative de la plateforme a été menée dans le cadre de cette recherche de doctorat, néanmoins [dnarchi] a dès sa création été un projet collectif développé en collaboration avec un comité de

rédaction<sup>106</sup>. [dnarchi] est donc une plateforme gérée par un comité de rédaction. La mise en place d'un comité scientifique est en cours d'élaboration.

[dnarchi] est une plateforme dont les enjeux sont multiples :

- rendre public et visible les connaissances scientifiques construites sur la conception architecturale numérique aux praticiens et étudiants en architecture,

- participer aux débats scientifiques sur la conception architecturale numérique en constituant une communauté d'auteurs sur des thématiques spécifiques et en organisant leurs échanges par des séries d'articles ou des rencontres physiques.

- assister l'usage de la modélisation paramétrique en architecture par la participation à la constitution d'un espace des savoirs de la modélisation paramétrique,

En cela le but premier de [dnarchi] n'est pas d'être un support d'enseignement mais il permet de diffuser les savoirs qui participent à *l'espace des savoirs de la modélisation paramétrique*.

Finalement les supports didactiques [dnarchi] et [parametric-ressources] sont complémentaires et s'inscrivent dans des objectifs différents. On observe que, grâce à leur complémentarité, [dnarchi] et [parametric-ressources] répondent aux objectifs fixés en début de section (être disponible et utilisable en et hors de cours et, supporter l'apprentissage technique et théorique de la modélisation paramétrique) (cf. Tableau 40).

|   | Etre disponible pour une formation continue | Etre disponible et utilisable en cours |
|---|---|--|
| Supporter l'apprentissage par la diffusion de samples   |   | [parametric-ressources]                |
| Supporter l'apprentissage en participant à la construction d'un espace de la modélisation paramétrique en conception architecturale | [dnarchi]                                   |  |

**Tableau 40 : Objectifs respectifs des plateformes [dnarchi] et [parametric-ressources]**

<sup>106</sup> Le comité de rédaction de septembre 2011 à septembre 2012 est composé de : Aurélie de Boissieu (reponsable éditoriale), Anne-Sophie Delaveau, Samia Ben Rajeb, Joaquim Sylvestre, François Guéna, Caroline Lecourtois et Thierry Ciblac. Le comité de rédaction de septembre 2012 à septembre 2013 est composé de Samia Ben Rajeb et Aurélie de Boissieu (responsables éditoriales) et de Marion Roussel et Anne-Sophie Delaveau.

### **8.3 Support pédagogique, pour un enseignement de la modélisation paramétrique en conception architecturale**

Dans cette section nous proposons un programme pédagogique à partir des objectifs précédemment définis (cf. 8.1.2.4, p.269). Nous proposons ici deux situations de transmissions des savoirs, savoir-faire et compétences de la modélisation paramétrique en conception architecturale :

- une pédagogie par travaux dirigés (8.3.2),
- une pédagogie par projet (8.3.3).

#### **8.3.1 Inscription d'un enseignement de la modélisation paramétrique dans les cursus d'enseignements de l'architecture existants**

##### **- Formations existantes à la modélisation paramétrique**

La formation à la modélisation paramétrique est présente à l'étranger dans les formations initiales des architectes, particulièrement dans les écoles où des centres de recherches s'intéressent aux enjeux du numérique comme aux Etats-Unis, en Angleterre, à Ghent en Belgique, Vienne en Autriche, Melbourne en Australie *etc.*

En France, et particulièrement en Ecole Nationale Supérieure d'Architecture (ENSA), la formation à la modélisation paramétrique est de plus en plus présente au travers de l'enseignement du logiciel Grasshopper (cf. 2.3.1), depuis ces trois dernières années. Dans les formations existantes en ENSA, on peut distinguer actuellement deux modes d'enseignements de la modélisation paramétrique : - un enseignement par le projet, dans le cadre d'ateliers (ou Studio) courts ou longs (comme à l'ENSA de Lyon, l'ENSA de Versailles, l'ENSA de Paris la Villette ou l'ENSA de Paris Malaquais) et; - un enseignement par travaux dirigés (comme à l'ENSA de Paris la Villette et l'ENSA de Marne la Vallée). Ces enseignements s'inscrivent dans les champs « Sciences et Techniques pour l'Architecture », « Théorie et Pratique de la Conception Architecturale et Urbaine » ou « Techniques de Représentation ». Certains cours sont également proposés hors ENSA en e-Learning (tels que proposés par Bentley System pour Generative Components) ou dans des structures de formations professionnelles.

En ENSA, la plupart des formations aux technologies numériques s'inscrivent dans le champ « Techniques de Représentation » (Lecourtois 2008). Cependant, nous avons vu précédemment l'importance des savoirs fondamentaux de géométrie et d'informatique dans l'enseignement que nous proposons (cf. 8.1.2). C'est pourquoi, dans le cadre d'une inscription en ENSA, l'enseignement que nous proposons s'inscrirait plutôt dans le champ « Sciences et Techniques pour l'Architecture ».

### - Inscription de l'enseignement dans le cursus de l'étudiant

Actuellement dans les ENSA, le cursus des étudiants est organisé en deux cycles : le cycle licence et le cycle master [culturecommunication.gouv.fr]. Le premier cycle est dédié à une formation commune et le deuxième cycle est le lieu d'une spécialisation de l'enseignement, choisie par l'étudiant parmi les possibles ouverts par l'école (Violeau 1999) [culturecommunication.gouv.fr].

Etant donné l'importance des savoirs nécessaires à l'usage de la modélisation paramétrique en conception architecturale, il ne semble pas pertinent d'inscrire un enseignement de la modélisation paramétrique trop tôt dans le cursus de l'étudiant.

Nous reprenons le point de vue de Ciriani lorsqu'il décrit le premier cycle de la formation en architecture comme tout à la fois « une synthèse », un « avant-poste » et une « introduction » (Ciriani in Violeau 1999, p.22). Dans ce cadre et étant donné les enjeux de l'usage d'outils numériques aujourd'hui (cf. Chapitre 3), il nous semble indispensable d'inscrire en premier cycle :

- une formation obligatoire à la géométrie (comme c'est déjà le cas en ENSA);
- ainsi qu'une formation obligatoire aux enjeux du numérique, inscrits dans le cadre de cours d'histoire de l'architecture contemporaine ou encore d'histoire des techniques.

Ces enseignements sont indispensables à un apprentissage éclairé de la modélisation paramétrique en conception architecturale. Ils permettent à l'étudiant l'assimilation de savoirs permettant d'aiguiser son esprit critique par rapport aux divers usages possibles du numérique et de poser quelques prérequis en géométrie.

|            |    |  |
|------------|----|--|
| 1er cycle  | L1 |  |
|            | L2 | Enseignement obligatoire de la géométrie // tronc commun   |
|            | L3 | Enseignement obligatoire des enjeux du numérique // tronc commun   |
| 2eme cycle | M1 | Enseignement de la MP par TD niveau 1 // spécialisation<br>Enseignement de la MP par le projet // spécialisation |
|            | M2 | Enseignement de la MP par TD niveau 2 // spécialisation  |

**Tableau 41 : Inscription de l'enseignement de la modélisation paramétrique dans le cursus d'un étudiant en ENSA**

L'enseignement de la modélisation paramétrique que nous proposons est un approfondissement de ces premiers cours. Nous proposons ici deux enseignements de la modélisation paramétrique : l'un par travaux dirigés, l'autre par projets. Ces enseignements peuvent être obligatoires ou optionnels en fonction de la politique de spécialisation de l'école (cf. Tableau 41).

### **8.3.2 Pédagogie par Travaux Dirigés**

Le terme « pédagogie » désigne l'articulation entre les processus de transmission de savoirs par l'enseignant et d'acquisition de ces savoirs par l'apprenant (Meirieu 2004; Altet 2006, p.16). Nous proposons ici de décrire une situation favorable à l'apprentissage de l'usage de la modélisation paramétrique par les architectes en vue de répondre aux objectifs pédagogiques précédemment formulés (cf. 8.1.2.4).

Pour définir notre programme pédagogique, nous décrivons dans cette section :

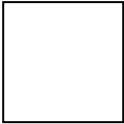
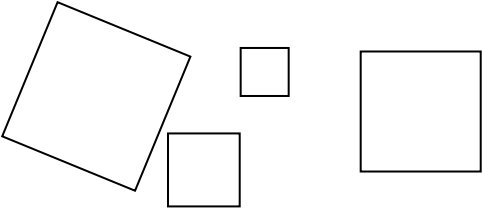
- la structure générique des séquences d'apprentissage mises en place,
- le contenu des séquences d'apprentissage,
- la mise en œuvre des séquences.

Dans cette section, nous nommons « séances » un temps de cours et « séquence » l'organisation de plusieurs activités avec un début, une fin et un objectif. Une séance peut être composée de plusieurs séquences.

#### **8.3.2.1 Définition des séquences d'apprentissage en deux temps: de l'objectivation au réinvestissement**

##### **- Une démarche d'apprentissage constructiviste : par l'exemple et par l'exercice**

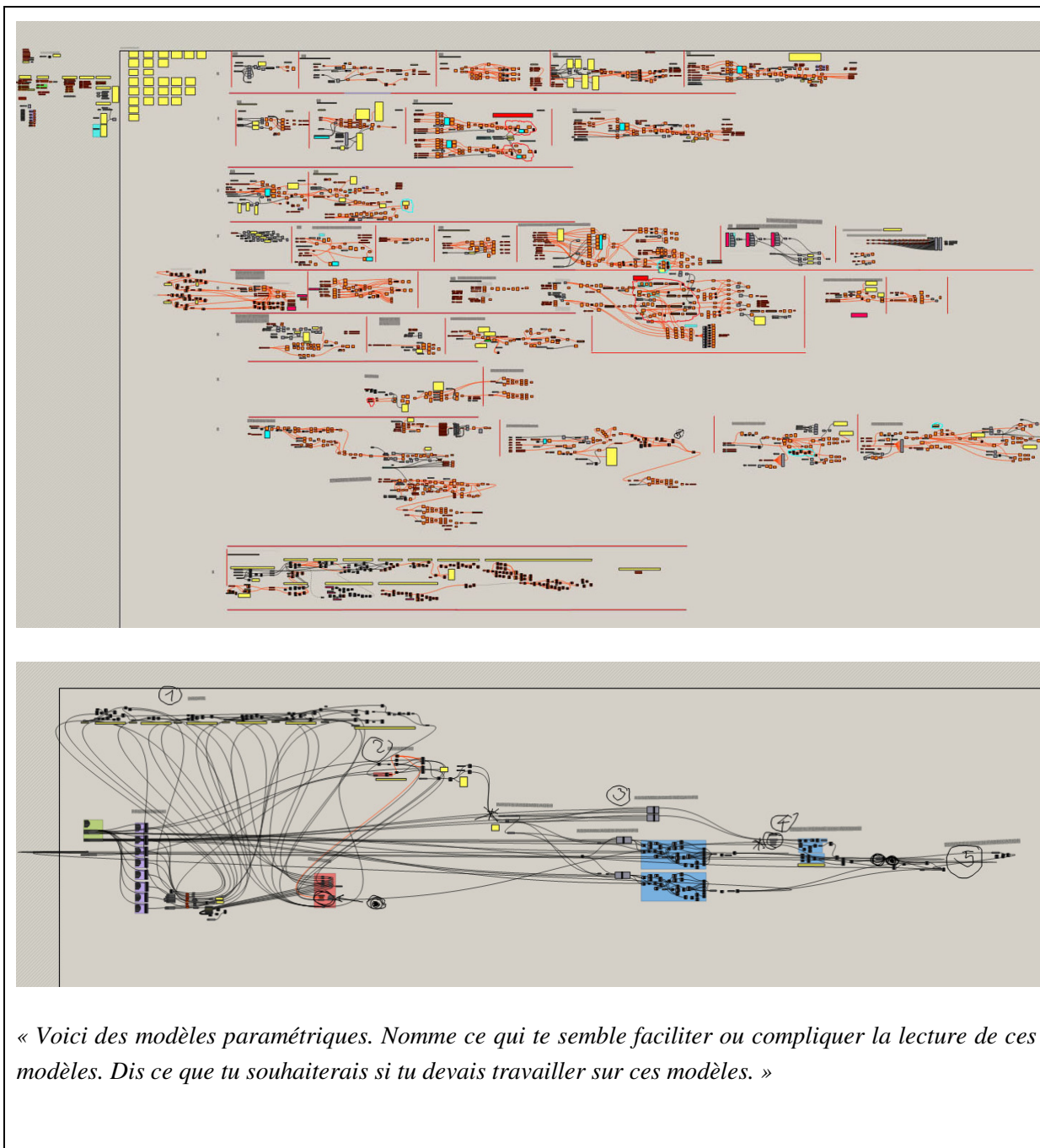
Pour Meirieu, lors d'une situation d'apprentissage en interaction avec un enseignant, l'important est l'activité mentale effectuée de l'étudiant. Pour favoriser cette activité mentale, Meirieu revendique l'usage de démarches dites constructivistes, en opposition aux démarches imposives (cf. Tableau 42). Les démarches constructivistes, comme l'induction par exemple, permettent à l'étudiant de construire des savoirs à partir de situations, notions ou exercices proposés par l'enseignant. C'est cette démarche, basée sur l'utilisation d'exemples et d'exercices, que nous reprenons pour un enseignement de la modélisation paramétrique.

| Démarche impositive  | Démarche constructiviste  |
|--|---|
| <p data-bbox="188 322 606 353">« Voici un carré, observe le bien. »</p>  <p data-bbox="188 645 775 676">« Dis quelles sont les caractéristiques du carré. »</p> | <p data-bbox="807 322 1056 353">« Voici des figures. »</p>  <p data-bbox="807 645 1394 721">« Trouve ce que ces figures ont en commun »<br/>(induction)</p> |

**Tableau 42: Démarches imposives et constructiviste d'après (Meirieu 20014)**

Par exemple, dans le cas d'un cours avec pour objectif « organiser un modèle paramétrique », on peut mettre en place une séquence d'apprentissage constructiviste en présentant plusieurs cas de modèles paramétriques plus ou moins organisés (cf. Tableau 43). Par comparaison ou projection (« Si vous deviez travailler sur ces modèles, comment vous sentiriez vous ? ») l'étudiant peut évaluer les points importants de l'organisation du modèle paramétrique le plus lisible, par rapport aux difficultés posées par l'autre modèle. Les ressentis et les observations des étudiants peuvent être formulés en commun, discutés et débattus. L'enseignant peut ensuite, à partir des diverses propositions, expliciter les points importants de l'organisation d'un modèle. Pour Grasshopper, ces points peuvent être: - grouper, structurer les composants, - nommer et commenter les étapes d'un modèle, - identifier les entrées et sorties de ces étapes.





**Tableau 43: Exemple de mise en place d'une démarche constructiviste pour l'apprentissage de la modélisation paramétrique**

### **-Proposition de scénario pédagogique**

Pour l'apprentissage de la modélisation paramétrique, des séances de 3 à 4h permettent de mettre en place deux temps : 1-un temps de démonstration de savoirs propres à la modélisation paramétrique dans une démarche constructiviste et, 2- un temps d'appropriation de ces savoirs par un réinvestissement de ceux-ci dans de nouveaux exercices (cf. Figure 125). Le temps 1 correspond à l'explicitation des objectifs de la séquence par l'enseignant (en termes de savoirs et de questions techniques abordées), ainsi qu'à la réalisation en commun d'un premier modèle paramétrique. Le

temps 2 correspond à un travail de réinvestissement du savoir par l'étudiant dans un transfert vers un nouvel exercice.

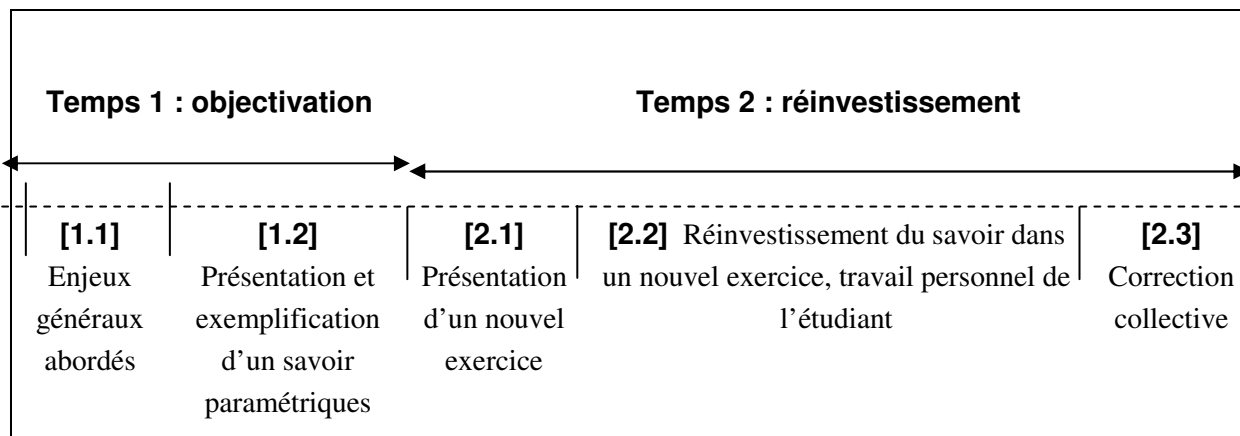


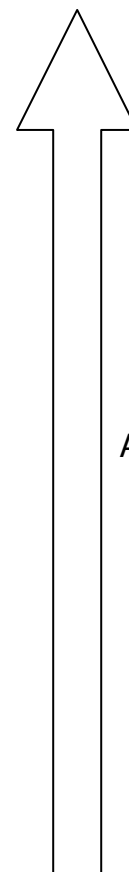
Figure 125 : Structure d'une séquence d'apprentissage de la modélisation paramétrique pour les architectes

### -Supports didactiques et processus d'autonomisation

En début de séquence d'apprentissage, l'étudiant est dépendant de l'enseignant qui le guide. L'étudiant est accompagné et exerce sa nouvelle habileté dans une situation imposée (l'exercice). Au fur et à mesure de la séquence d'apprentissage, les aides prodiguées à l'étudiant doivent pouvoir s'alléger, jusqu'à ce que l'étudiant puisse remplir une tâche qu'il a lui-même défini (un modèle paramétrique pour un projet architectural qui lui est propre) en mettant en œuvre seul sa nouvelle habileté. La séquence d'apprentissage doit pouvoir fonctionner comme un processus d'autonomisation.

Les supports didactiques peuvent aider l'étudiant à s'autonomiser car ils accompagnent ce processus d'autonomisation, en particulier en permettant à l'étudiant d'avoir accès à des ressources structurées sur l'espace des savoirs de la modélisation paramétrique (cf. [dnarchi] et [parametric-ressources]). Certains supports sont propres à l'élaboration du temps 1 et servent à l'enseignant pour mettre en place une démarche de construction des notions. Ces supports sont des exposés, des images de projets, des annotations au tableau, etc. D'autres supports visent à être sollicités à tout moment comme aide à la modélisation paramétrique (cf. Figure 20). Ces supports permettent à la fois un accès à : 1-des connaissances techniques propres au logiciel utilisé, c'est ce que nous proposons avec le développement de la plateforme [parametric-ressources] (cf. 8.2.1) et, 2-des connaissances théoriques plus générales sur la modélisation paramétrique, la géométrie ou le numérique, c'est ce que nous proposons avec le développement de la plateforme [dnarchi] (cf. 8.2.2).

|               | <i>Description de la séquence</i>   | <i>Recours à [parametric-ressources]</i>                         |
|---------------|---|--|
| Temps [2 bis] | L'étudiant met en œuvre une habileté dans un cas spécifique de modélisation paramétrique.<br>Il s'exerce dans un contexte qu'il définit lui-même.   | [parametric-ressources] sert de support à la mémoire             |
| Temps [2]     | L'étudiant met en œuvre une habileté dans une nouvelle tâche, en vue de s'approprier une habileté fraîchement acquise.<br>Il s'exerce dans des situations définies et normalisées par l'enseignant. | [parametric-ressources] sert d'exemplification de cas différents |
| Temps [1]     | L'enseignant guide l'étudiant dans un processus mental par des questions, des successions de consignes, etc.  | [parametric-ressources] sert de support d'illustration           |



Autonomisation de l'étudiant

Figure 126 : Processus d'autonomisation et recours aux supports didactiques

**-exemple de séquence détaillée**

Par exemple, une séquence de cours portant sur « évaluer l'ensoleillement d'un espace » dans Grasshopper pourrait s'organiser comme ceci :

|   |
|---|
| <b>SEQUENCE « évaluer l'ensoleillement d'un espace<br/>grâce à la modélisation paramétrique »</b> |
| <b>TEMPS 1</b>  |

### **[1.1] Expliciter les enjeux généraux abordés :**

- Expliciter l'intérêt de l'évaluation en modélisation paramétrique
- Présentation rapide de techniques existantes liées à l'objet du cours: présentations de moteurs d'évaluation comme ecotech, présentation rapide de la modélisation inverse, référence à un article de [dnarchi]
- Rappel des principes de géométrie des projections qui seront utilisés, utilisation d'un article de [dnarchi]
- Présentation des patterns Projector et Reporter en interrogeant les stratégies possibles dans un modèle paramétrique : « *Si l'on devait modéliser une façade pour en évaluer l'ensoleillement, comment organiseriez-vous votre modèle ? à votre avis, dans cette organisation, qu'est-ce qui peut être générique ? qu'est ce qui est propre à l'évaluation de l'ensoleillement ?* »

### **[1.2] Présentation du processus de modélisation paramétrique propre au modèleur étudié et application dans un modèle construit en cours**

- utilisation du composant « mesh shadow » de Grasshopper dans un exemple simple  
« *on a besoin d'un composant permettant de projeter l'ombre d'un volume, quel pourrait-être le composant adapté ? il pourrait être dans quel menu de l'interface ? D'après vous, de quoi a-t-il besoin comme données d'entrée ? Pourquoi le composant ne fonctionne-t-il pas ?* »
- présentation de samples appliquant le pattern projector sur [parametric-ressources]

## **TEMPS 2**

### **[2.1] Explication de l'exercice demandé**

L'enseignant fournit un site modélisé sur Rhinocéros et des exigences quant à la conception d'un moucharabieh et à sa performance du point de vue de l'ensoleillement (par exemple, l'ensoleillement doit être de 40 % à midi en hiver à l'intérieur de l'espace).

### **[2.2] Travail personnel de l'étudiant**

L'étudiant tente par lui-même de remplir la nouvelle tâche.

Après un premier temps où l'étudiant s'approprie seul la tâche, l'enseignant peut aider ponctuellement l'étudiant sur un problème précis.

Il peut consulter les samples publiés sur [parametric-ressources] et relatifs aux questions abordées

### **[2.3] Correction collective**

En fin de séquence, l'enseignant évalue si la nouvelle tâche a été remplie et montre le processus de réalisation de la tâche : « *comment avez-vous fait pour... ? Est-ce que quelqu'un a essayé de ... ? Ce que je vous propose pour ... c'est de ....* »

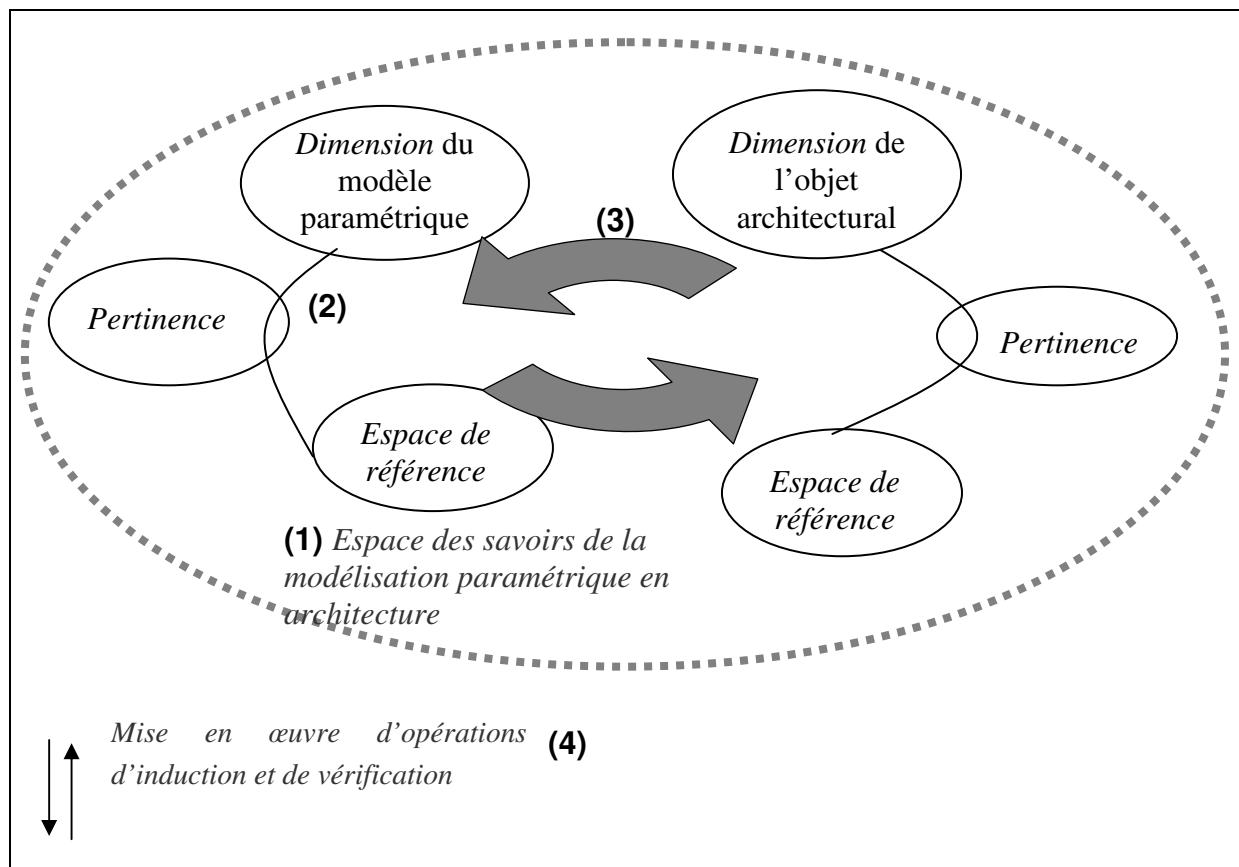
**Tableau 44: Exemple de séquence de cours pour l'apprentissage de la modélisation paramétrique**

Dans cet exemple de séquence, on s'aperçoit que le savoir abordé est technique et lié à des questions récurrentes en architecture. Cependant, les opérations de conception de modèles paramétriques mises en œuvre lors de cette séquence peuvent facilement partager des références et des *pertinences* (cf. 6.3.2 p.214) d'opérations de conception architecturale dans des cas d'*échelles* géographique ou technique par exemple.

Dans cette organisation de séquence en deux temps, des mécanismes et des savoirs généraux sont abordés concrètement au travers de la réalisation de modèles paramétriques spécifiques. On reprend ici ce qui était observé lors de l'usage des samples : la réalisation ou la manipulation de modèles paramétriques est un médium à l'apprentissage de savoirs particuliers. L'organisation de la séquence autour de la réalisation, collective et personnelle, de modèles spécifiques lors d'exercice permet donc aux apprenants de construire certains savoirs par eux-mêmes plutôt que de les recevoir déjà élaborées, mais aussi de cultiver un savoir-être lié à l'esprit d'initiative et de travailler une aptitude à chercher les informations appropriées pour répondre à un problème nouveau.

### **8.3.2.2 Contenus des séquences d'apprentissage**

Nous avons précédemment établi un scénario pédagogique en fonction du processus d'apprentissage visé. Dans cette section, nous proposons quelques exemples de contenus de séquences d'apprentissage permettant d'aborder et d'acquérir les ressources nécessaires mises en œuvre lors de la conception de modèles paramétriques en conception architecturale, telles qu'elles ont été explicitées précédemment (cf. Figure 127).



**Figure 127 : Schématisation de l'activité cognitive mise en œuvre lors de la conception de modèles paramétriques en architecture et des différents points à aborder lors d'un apprentissage de cette activité**

Dans le cadre d'un enseignement de la modélisation paramétrique en conception architecturale, nous avons identifié un certain nombre d'objectifs pédagogiques que l'on peut rapprocher:

- (1) la construction d'un espace des savoirs de la modélisation paramétrique en conception architecturale ;
- (2) l'apprentissage de la mise en œuvre d'opérations de conception de modèles paramétriques (découpage, référencement et dimensionnement) ;
- (3) l'apprentissage de l'activité de traduction d'intentions architecturales en géométrie paramétrique ;
- (4) et enfin de l'apprentissage de mise en œuvre d'opérations logiques d'induction et de vérification.

Ces quatre points sont représentés dans la Figure 127 et nous permettent de décrire les contenus des séquences pédagogiques que nous proposons dans les paragraphes suivants.

#### **- Comprendre le rapport entre Rhinocéros et Grasshopper**

Lors de cette séquence, l'étudiant fait des actions telles que - Importer une géométrie depuis Rhinocéros vers GH ; - exporter une géométrie depuis GH vers Rhinocéros ; - exporter une géométrie

depuis GH vers Rhinocéros en automatisant les calques de destination en vue de produire un document propre à être utilisé avec une découpe laser.

Cette séquence permet d'aborder concrètement la question du plugin et du rapport de l'instance au modèle paramétrique. Elle participe donc à la construction d'un *espace des savoirs de la modélisation paramétrique* (cf. Tableau 37, point (1)) et au développement d'un savoir-faire lié à la mise en œuvre des opérations logiques *d'induction* et de *vérification* (cf. Tableau 37, point (2)).

#### **- Comprendre la méthode «Divide and conquer» et l'appliquer à la conception d'un motif sur une surface suivant différentes stratégies**

Lors de cette séquence, l'étudiant construit plusieurs modèles lui permettant de répéter un module sur une surface prédéfinie. Ainsi, sont abordés – la construction d'un maillage à partir de la subdivision d'une surface de référence ; - la construction d'un « tissage » à partir d'une trame de points générée sur la surface ; et enfin la répétition d'un motif par « morphing » d'une forme de départ sur la surface subdivisée.

Lors de cette séance, l'étudiant expérimente la possibilité d'obtenir une même géométrie mais des domaines de solutions différents à partir de modèles paramétriques différents. En cela, la séquence participe à la construction d'un savoir-faire propre à la mise en œuvre d'opérations de conception de modèles paramétriques (cf. Tableau 37, point (2)). Par ailleurs l'étudiant expérimente aussi la possibilité de faire évoluer un modèle, ici pour utiliser une même surface et des motifs différents pour explorer différents domaines de solutions. La séquence permet à l'étudiant de se construire un savoir-faire propre à la mise en œuvre d'opérations de conception de modèles paramétriques successives (cf. Tableau 37, point (2)) en particulier en exemplifiant la stratégie « Divide and Conquer » formalisée par Woodbury et participant d'un *espace des savoirs de la modélisation paramétrique* (cf. Tableau 37, point (1)).

Par ailleurs, les trois stratégies de création de motif présentées appliquent chacune un pattern spécifique, respectivement le pattern *place holder*, le pattern *increment* et le pattern *Jig*.

#### **- Evaluation et optimisation / fabrication**

Lors de cette séquence, l'étudiant modélise une structure triangulée simple dont il identifie les contraintes et les variables. Puis, il évalue les dimensions des barres en vue de comprendre les difficultés constructives posées par la géométrie du modèle. La géométrie du modèle est ensuite optimisée pour pouvoir standardiser les éléments constructifs. Pour cela, on calcule l'écart type entre les dimensions des éléments de structure, puis on utilise le moteur d'optimisation de GH « Galapagos ».

Lors de cette séquence, l'étudiant met en œuvre une logique d'évaluation puis d'optimisation de la géométrie conçue. Ces stratégies correspondent aux patterns *reporter* et *goal seeker*. La séquence participe ainsi à la construction de *l'espace des savoirs de la modélisation paramétrique* de l'étudiant (cf. Tableau 37, point (1)).

#### **- Récupérer et envoyer des informations depuis Grasshopper vers d'autres applications**

Lors de cette séquence, l'étudiant utilise le plugin de GH « Firefly » et une carte Arduino pour récupérer les données issues d'une captation (en fonction des capteurs disponibles) dans un modèle

géométrique. Le plugin de GH « Mosquitos » est également utilisé pour récupérer dans Grasshopper des données issues du réseau social twitter, ainsi que le composant « read file » de Grasshopper.

Lors de cette séance, l'étudiant utilise des plugins de Grasshopper, qui permet l'ouverture des fonctionnalités proposées sur le modeleur, mais aussi l'ouverture possible d'un modèle paramétrique grâce par exemple à la captation ou à l'open data. Cela participe à la construction de *l'espace des savoirs de la modélisation paramétrique* de l'étudiant (cf. Tableau 37, point (1)).

Ces séquences décrivent la mise en œuvre de questions techniques précises (comme optimiser une structure, créer un module de façade réactif à des données de l'environnement, etc.). Lors de ces séquences, des savoir-faire transversaux sont également abordés. Ils le sont en lien avec l'architecture abordée par la séquence. Ces savoir-faire transversaux peuvent également être abordés comme une parenthèse au cas où des étudiants rencontrent des difficultés. Ces savoir-faire transversaux sont décrits dans les paragraphes suivant.

#### **-débugger**

Quand un étudiant se retrouve face à un modèle ne fonctionnant pas ou ne donnant pas le résultat escompté, l'étudiant doit pouvoir mettre en place des stratégies d'exploration et de vérification de son modèle en vue de résoudre le(s) problème(s) rencontrés. Ce savoir-faire est abordé tout au long de la formation lors des différentes réalisations de modèles paramétriques. Il peut également être l'objet d'un exercice ou même d'un test particulier, lors duquel un modèle ne fonctionnant pas est fourni à l'étudiant, ce dernier doit alors identifier le problème et le résoudre. Cette compétence est indispensable à la mise en œuvre d'opérations de conception du modèle paramétrique (cf. Tableau 37, point (2)).

#### **- Utiliser un nouveau composant**

Quand un étudiant se retrouve devant une nouvelle fonction ou devant un nouveau composant (dans le cas de Grasshopper), il doit pouvoir mettre en place des stratégies d'exploration et avoir suffisamment de connaissances du modeleur pour comprendre et s'appropriier le composant. Ces stratégies et ce savoir-faire sont continuellement mis en œuvre lors des séquences puisque de nouvelles fonctions sont intégrées pour chacun des problèmes techniques abordés. Néanmoins, un exercice ou même un test peut être proposé pour évaluer ce savoir-faire : on soumet à l'étudiant un composant précis, jamais vu en cours, et qu'il doit mettre en œuvre pour obtenir un résultat défini par l'enseignant. Comme précédemment, cette compétence est indispensable à la mise en œuvre d'opérations de conception du modèle paramétrique (cf. Tableau 37, point (2)).

#### **- S'appropriier un modèle développé par un autre**

Un étudiant doit pouvoir comprendre et s'appropriier un modèle paramétrique qu'il n'a pas lui-même développé. Cela est indispensable pour pouvoir débbugger un modèle et pour pouvoir utiliser la méthode des patterns. Ce savoir-faire s'acquiert comme celui du debugging. C'est une logique d'exploration. Ce savoir-faire peut faire l'objet d'un exercice ou d'un test lors duquel un modèle est



fourni à l'étudiant, une consigne précise à l'étudiant la modification qui est demandée (des cercles à la place des carrés, liés le rayon d'un cylindre à sa longueur, etc.).

### **8.3.2.3 Evaluation de l'enseignement**

Pour cet enseignement de la modélisation paramétrique par travaux dirigés, nous proposons trois modes d'évaluation complémentaires :

-le contrôle continu des exercices faits en cours ;-des tests ponctuels, courts, sur des points précis (debugger un modèle, modifier un modèle que l'on n'a pas développé soi-même, etc.) et; -un examen final consistant en la réalisation d'un projet d'architecture court sur le modèle paramétrique étudié.

#### **-contrôle continu**

Lors des séquences, certains exercices sont faits en commun en début de séquences, d'autres sont réalisés par les apprenants dans un objectif de réinvestissement des savoirs abordés. Ces exercices peuvent être contrôlés pour évaluer l'assiduité des étudiants ainsi que leur assimilation des différentes séquences.

#### **-examen final**

L'examen final proposé est le même exercice que celui construit pour les expérimentations. Les apprenants sont en binômes pour construire un modèle paramétrique propre à un projet d'architecture qu'ils construisent en même temps.

#### **-tests ponctuels**

Dans les tests ponctuels, nous proposons par exemple :

- un test lors duquel l'apprenant doit identifier les principales règles de génération d'un modèle à partir de cas particuliers simples ;

- un test lors duquel l'apprenant doit modifier un modèle paramétrique qu'il n'a pas lui-même développé et l'adapter à des objectifs de modélisation donnés par l'enseignant ;

-un test lors duquel l'apprenant reçoit un modèle paramétrique qu'il n'a pas lui-même développé, qui ne fonctionne pas, et qu'il doit « debugger » (par exemple en adaptant les structures de données du modèle pour le cas d'un test de niveau avancé).

Ces trois modes d'évaluation permettent d'interroger la réalisation de tous les objectifs pédagogiques précédemment définis (cf. Tableau 45).

| <i>Objectifs pédagogiques</i>   | <i>Mode d'évaluation</i>                                      |
|---|---|
| -l'apprenant doit savoir où trouver une information s'il se trouve dans une situation inconnue ou devant une question technique (propre au modeleur ou propre à la géométrie ou à l'informatique en général) qu'il ne maîtrise pas.   | Par le projet (examen final)                                  |
| -l'apprenant doit maîtriser des bases de géométrie paramétrique (définir une surface développable, un plan, une courbe, etc.) et d'informatique (boucle, plugin)  | Examen final et contrôle continu des exercices faits en cours |
| -l'apprenant doit posséder des repères sur les principaux enjeux liés au numérique faisant l'actualité de leur futur cadre de travail   | Par le projet (examen final)                                  |
| -l'apprenant doit pouvoir choisir entre les principaux modeleurs paramétriques et leurs modes d'échanges avec les principaux moteurs d'évaluation, d'interaction et de fabrication, pour choisir le logiciel, ou l'agencement de logiciels, le plus approprié à son projet. | Par le projet (examen final)                                  |
| -l'apprenant doit pouvoir manipuler le modeleur paramétrique abordé lors de la formation pour concevoir un modèle paramétrique simple   | Contrôle continu  |
| -l'apprenant doit pouvoir appliquer la stratégie « Divide and conquer » à la conception d'un modèle paramétrique simple   | Contrôle continu  |
| -l'apprenant doit pouvoir organiser et rendre lisible un modèle paramétrique qu'il a construit  | Contrôle continu  |
| -l'apprenant doit pouvoir lire un modèle paramétrique qu'il n'a pas lui-même développé et le modifier pour l'adapter aux objectifs de modélisation qui lui sont personnels  | Test ponctuel   |
| -l'apprenant doit pouvoir interpréter une intention architecturale simple en géométrie paramétrique   | Par le projet (examen final)                                  |

|   |                                  |
|---|----------------------------------|
| -l'apprenant doit pouvoir organiser son modèle paramétrique en anticipant de futures évolutions de celui-ci   | Contrôle continu                 |
| l'apprenant doit pouvoir organiser son travail, mais aussi écouter et interpréter un propos de façon à travailler en équipe   | Par le projet (examen final)     |
| -décomposer une intention de modèle paramétrique en termes d'enchaînements logiques et de dépendances géométriques,   | Par le projet (examen final)     |
| -il doit pouvoir dimensionner un modèle paramétrique en fonction de pertinences qui lui sont propres, en particulier en fonction de pertinences liées à la conception architecturale.   | Examen final et contrôle continu |
| -l'apprenant doit pouvoir tester la viabilité d'un modèle paramétrique simple en testant les extremums de son domaine de solutions  | Contrôle continu                 |
| -l'apprenant doit pouvoir identifier les principales règles de génération d'un modèle à partir de cas particuliers simples  | Test ponctuel                    |
| -l'apprenant doit connaître des principaux enjeux techniques liés à la modélisation paramétrique : évaluation structurelle, évaluation des caractéristiques géométriques d'une forme (développable ? plane ? constructible avec des modules standardisés ?), évaluation de son ensoleillement, évaluation de caractéristiques quantitative (surface constructible ? quantité d'acier dans une charpente ? etc.) | Contrôle continu                 |

**Tableau 45 : Mode d'évaluation des objectifs pédagogiques précédemment définis**

Cet enseignement de la modélisation paramétrique en architecture par travaux dirigés est efficace du point de vue de l'apprentissage de la mise en œuvre d'opérations de conception de modèles paramétrique et de la construction d'un espace des savoirs de la modélisation paramétrique. Ces compétences et ces savoirs étant difficiles à acquérir, un tel enseignement focalisé sur la mise en œuvre de techniques se justifie. Néanmoins, elle permet peu d'aborder la question de l'entremêlement entre conception architecturale et conception de modèle paramétrique dans le cas de projets personnels. C'est pour cela qu'un enseignement de la modélisation paramétrique par travaux dirigés est complémentaire à un enseignement par le projet.

### 8.3.3 Pédagogie par projet

Nous nommons pédagogie **par** projet le programme d'apprentissage que nous proposons et pour lequel nous nous appuyons sur les apports de la pédagogie dite « **de** projet » (altet p.68 et 71)). Dans cette section nous entendons le terme « projet » dans son sens pédagogique générique d'entreprise visant à la réalisation par un apprenant d'une production concrète en intégrant des savoirs et savoir-faire nouveaux (Altet 2006) et non dans son sens restreint de « projet d'architecture » tel qu'il est usuellement utilisé en école d'architecture.

L'enseignement de la modélisation paramétrique par projet que nous proposons vise à permettre à l'apprenant d'acquérir des compétences spécifiques à l'entremêlement entre conception architecturale et conception de modèles paramétriques. Ces compétences sont liées en particulier à l'opération de *traduction en géométrie paramétrique*.

Par rapport à la pédagogie par travaux dirigés précédemment proposée, cet apprentissage par projet vise donc de la même manière l'apprentissage de la modélisation paramétrique en conception architecturale et permet un approfondissement de l'apprentissage de la mise en œuvre de liens entre modélisation paramétrique et conception architecturale. Cet enseignement **par** le projet vise donc à enrichir les compétences de l'apprenant **pour** le projet d'architecture.

Cette section présente un scénario pédagogique générique (8.3.3.2) ainsi que des contenus (8.3.3.3) et des modalités de mises en œuvre du scénario (8.3.3.4) adaptés à un enseignement de la modélisation paramétrique en conception architecturale par projet.

### 8.3.3.1 Objectifs de l'enseignement de la modélisation paramétrique en conception architecturale par projet

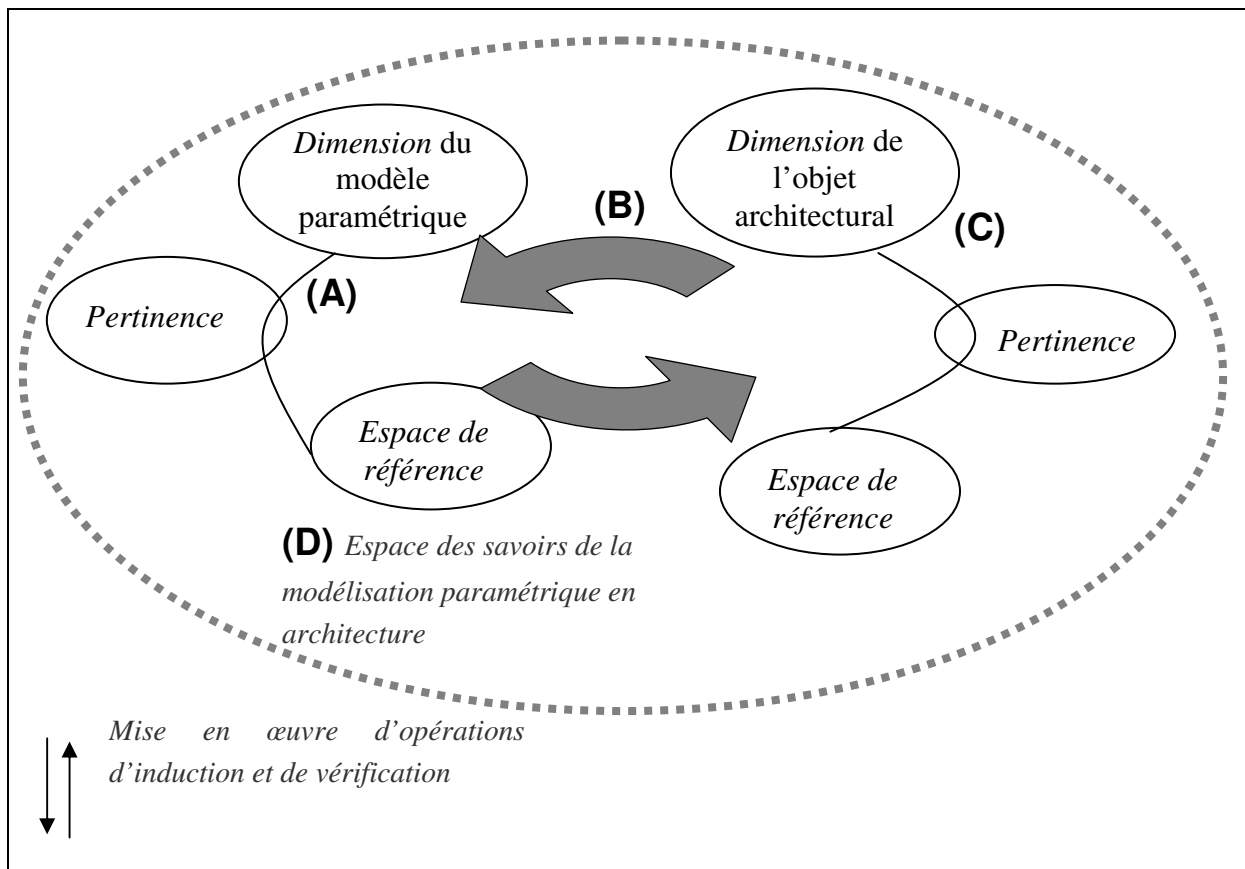


Figure 128 : Schématisation de l'activité cognitive mise en œuvre lors de la conception de modèles paramétriques en architecture et des différents points à aborder lors d'un apprentissage de cette activité

Par rapport à l'enseignement par travaux dirigés précédemment proposé, l'enseignement par projet de la modélisation paramétrique en conception architecturale met l'accent sur (cf. Figure 129):

- A- l'apprentissage de la mise en œuvre d'opérations de conception de modèles paramétriques (*découpage, référenciation et dimensionnement*) ;
- B- l'apprentissage de l'activité de traduction d'intentions architecturales en géométrie paramétrique ;
- C- l'apprentissage de la mise en œuvre d'opérations de conception architecturale (*découpage, référenciation, dimensionnement*) ;
- D- la construction d'un espace des savoirs de la modélisation paramétrique en conception architecturale.

Parmi les objectifs précédemment identifiés (cf. 8.1.2.4 p. 269), cet apprentissage par projet de la modélisation paramétrique relève donc particulièrement des objectifs suivant :

-A la fin de la formation, l'apprenant doit pouvoir dimensionner un modèle paramétrique en fonction de *pertinences* qui lui sont propres, en particulier en fonction de *pertinences* liées à la conception architecturale (cf. Figure 129 A et B).

- A la fin de la formation, l'apprenant doit pouvoir décomposer une intention de modèle paramétrique en termes d'enchaînements logiques et de dépendances géométriques (cf. Figure 129 A),

- A la fin de la formation, l'apprenant doit pouvoir organiser son travail, mais aussi écouter et interpréter un propos de façon à travailler en équipe (cf. Figure 129 D).

- A la fin de la formation, l'apprenant doit pouvoir trouver une information qui lui est nécessaire lorsqu'il est dans une situation inconnue ou devant une question technique (propre au modéleur ou propre à la géométrie ou à l'informatique en général) qu'il ne maîtrise pas (cf. Figure 129 D).

### 8.3.3.2 Scénario pédagogique

Pour répondre aux objectifs de l'enseignement et aux objectifs pédagogiques précédemment définis, nous nous appuyons sur les apports de la pédagogie de projet pour définir notre scénario pédagogique. En particulier, nous reprenons les sept caractéristiques proposés par Louis Legrand pour définir la pédagogie de projet (Legrand in altet p. 69-70). Ces caractéristiques sont explicitées dans les paragraphes suivants pour définir notre pédagogie de la modélisation paramétrique en architecture par projet.

#### **- l'implication affective et l'engagement personnel de l'apprenant:**

Pour qu'une pédagogie par projet soit efficace, l'implication personnelle de l'apprenant est majeure. L'adhésion à l'enseignement ainsi que le choix du projet doivent être libres et choisis<sup>107</sup>. En cela, le positionnement de l'enseignement que nous proposons dans le cycle master et dans le cadre d'enseignements choisis (optionnel ou projet choisi parmi plusieurs) (cf. 8.3.1) permet un investissement personnel de la part de l'apprenant. En effet, il semble qu'après trois ans de formation à la conception architecturale l'apprenant ait développé des intérêts et des curiosités qui lui sont propres (violeau 1999) et, si la modélisation paramétrique ou l'implication du numérique dans l'architecture en font parties, l'investissement de l'apprenant pourra être fort.

#### **- le recours au travail par équipe :**

Pour Legrand le travail d'équipe est indispensable à une pédagogie de projet (Legrand in Altet p.69). Il permet la répartition des tâches, la prise de décision collective et l'inscription du projet dans un contexte d'interactions social. Cet aspect de la pédagogie de projet parait tout particulièrement pertinent pour l'apprentissage de la modélisation paramétrique en conception architecturale. En effet, nous avons vu lors de l'état des pratiques de modélisation paramétrique en architecture (cf. chapitre 3) comme lors de l'analyse des opérations cognitives impliquées (cf. chapitre 6) que la collaboration entre experts de la modélisation paramétrique et acteurs de la conception architecturale était partie prenante de ces pratiques. Nous avons identifié que les

---

<sup>107</sup> Le sens de projet est ici entendu dans son sens général propre à la pédagogie et non dans son sens restreint de « projet d'architecture ». Si le projet d'architecture peut être imposé (au travers d'un programme, d'un site, etc) le projet de l'apprenant (explorer les usages, utiliser une référence précise, etc) lui est personnel et choisi.

opérations pragmatiques de *mise en commun* et d'*interprétation* intervenaient dans l'entremêlement entre modélisation paramétrique et conception architecturale. Une pédagogie par projet où les apprenants collaborent au sein d'équipe paraît donc particulièrement pertinente pour l'enseignement proposé.

**- la nécessité d'une planification collective du travail mené :**

Selon Legrand la planification du travail doit être effectuée collectivement et non imposé par l'enseignant. Entre les dates butoirs fixées par l'enseignant (comme les dates de début et de fin du projet par exemple), les apprenants doivent pouvoir planifier l'organisation de leur travail librement par des accords collectifs. Cette caractéristique semble tout à fait compatible avec une pédagogie de la modélisation paramétrique en conception architecturale par projet.

**- présentation du travail produit devant des personnes extérieures au projet :**

Pour Legrand, comme pour d'autres auteurs qui ont interrogés l'enseignement de la conception architecturale (Epron, Boudon, Toussaint) la visée de la présentation d'un projet devant un jury extérieur permet de stimuler l'apprenant en inscrivant l'apprentissage dans un contexte social valorisant. Pour cet auteur, le jury doit comporter des personnalités extérieures à l'enseignement. Dans le cas de notre enseignement cet aspect paraît pertinent parce qu'il permet de confronter les pratiques menées lors du projet à des acteurs pouvant les replacer dans une culture plus large ou à des spécialistes des sujets abordés.

**- la souplesse du phasage du projet :**

Lors du projet, l'apprenant doit pouvoir procéder par tâtonnements, essais et erreurs (GFEN in Altet p.73).

**\_ le recours à un travail personnel et à des décisions collectives :**

Dans une pédagogie par projet, si une grande place est faite à la prise de décision collective (sur la répartition des tâches, la planification du travail, les choix importants liés au projet, etc.) l'individu conserve un rôle majeure dans l'avancée du projet.

**- le positionnement de l'enseignant comme un régulateur et un informateur :**

Dans une pédagogie de projet, l'enseignant aide les apprenants à analyser les problèmes auxquels ils sont confrontés et fournit des informations complémentaires (GFEN in Altet p.74). L'enseignant est donc une personne ressource qui, sans s'imposer ni abandonner un groupe, intervient ponctuellement pour nourrir la pratique des apprenants.

Si toutes ces caractéristiques nous semblent pertinentes pour un apprentissage de la modélisation paramétrique en conception architecturale, il nous semble tout de même que de courtes interventions sur des points spécifiques à la technicité des modélisateurs paramétriques sont indispensables.

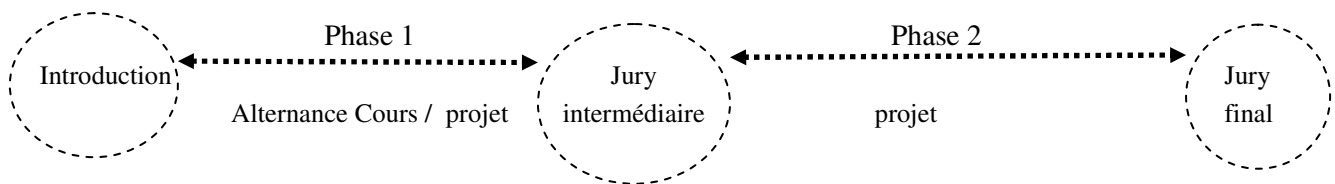
Nous proposons donc un atelier qui reprendrait les caractéristiques proposées par Legrand, tout en proposant au début de l'apprentissage de courtes sessions d'enseignement sur des savoirs et savoir-faire techniques. Sur un cours de dix séances, le scénario pédagogique s'organiserait comme suit (cf. Figure 129) :

### - phase 1 :

La phase 1 est initiée par une séance d'introduction et de présentation des objectifs du cours. Puis la phase 1 se constitue de séances partagées entre des enseignements spécifiques au projet et aux techniques utilisées (cours de modélisation paramétrique, conférence plus générale sur les enjeux du numérique, conférence de personnalités ayant une pratique liée au sujet, etc.) et, des séances d'atelier où les apprenants, organisés en groupes, développent leur projet. Cette phase se conclut par un jury intermédiaire qui vise à amener les apprenants à structurer leur projet après la première phase et, qui permet aux apprenants de prendre connaissance des projets des autres groupes et d'entendre les réactions du jury à leur sujet.

### - phase 2 :

La phase 2 fait suite au jury intermédiaire concluant la phase 1. Cette phase se constitue exclusivement de séances de projet où les groupes travaillent collectivement. Lors de cette phase les groupes sont accompagnés par les enseignants qui leur amènent des retours critiques et fournissent ponctuellement des informations complémentaires sur les références qui peuvent être pertinentes ou pour résoudre les problèmes techniques rencontrés. Cette phase se conclut par un jury final.



**Figure 129 : Proposition de scénario pédagogique pour un enseignement de la modélisation paramétrique en conception architecturale**

Ce scénario semble proche des scénarii fréquemment mis en œuvre lors d'ateliers visant l'apprentissage de compétences techniques et leurs mises en œuvre en conception architecturale. C'est par exemple le cas de la formation « Biodynamic Structures » organisée par l'Architecture Association à l'université de San Francisco en 2010, lors de laquelle des projets d'architectures « dynamiques » étaient développés dans le cadre de l'apprentissage de Grasshopper et d'Arduino. La formation se composait de dix jours. Les quatre premiers jours alternaient conférences, formations techniques et ateliers. Un jury intermédiaire a eu lieu le quatrième jour, il était suivi de quatre jours d'ateliers. La formation était conclue par un jury organisé sous la forme d'une exposition [sanfrancisco.aaschool].

### 8.3.3.3 Contenus de l'enseignement

Pour un enseignement de la modélisation paramétrique en conception architecturale par projet, les contenus spécifiques choisis permettent de focaliser l'apprentissage sur certains savoirs et de développer des savoir-faire transversaux liés aux compétences de mise en œuvre de ces savoirs.

A partir des contenus précédemment développés dans le cadre de la pédagogie par travaux dirigés que nous proposons et pour une complémentarité de nos deux enseignements, nous proposons trois



exemples de contenus d'enseignement de la modélisation paramétrique par projet, développés dans les paragraphes suivants.

**- « Lieux de convivialité, espace adaptable » Architectures dynamiques et performances bioclimatiques (Atelier Grasshopper, Arduino et Firefly)**

Un enseignement de la modélisation paramétrique en conception architecturale par projet pourrait avoir comme contenu l'enseignement de Grasshopper et d'Arduino en vue de développer des architectures dynamiques pouvant optimiser leurs performances bio-climatiques. Pour cela, le projet d'architecture et sa traduction au travers de modèles paramétriques font appel à des savoirs techniques liés à : -l'évaluation des performances énergétiques d'un bâtiment (ensoleillement, déperditions thermiques, etc.) ; -l'électronique et la possibilité de capter des informations du réel et d'intervenir sur le réel via des moteurs par exemple. Le projet fait également appel à des savoirs propres à la culture du numérique en général et du développement durable en architecture.

Dans le cadre de cet enseignement, nous proposons l'utilisation de Grasshopper pour le développement de modèles paramétriques, de cartes Arduino pour l'apprentissage de l'électronique et du plugin Firefly pour lier le modèle paramétrique et l'électronique développés pour le projet.

Arduino est une technologie visant à rendre accessible l'électronique aux débutants. Grâce aux cartes Arduino un apprenant peut rapidement contrôler des capteurs (de luminosités, de température, de poids, etc.) et des moteurs (moteurs pas à pas ou encore servo moteurs). Le plugin de Grasshopper nommé Firefly permet, quant à lui, de récupérer facilement les données captées par une carte Arduino et d'envoyer des données vers une carte et donc de contrôler un moteur (par exemple à une variable 0 ou 1 correspondent les positions marche et arrêt d'un moteur).

Un exemple de sujet pour cet enseignement est proposé ci-dessous.

Sujet du projet « Espace de convivialité adaptable »,

Architectures dynamiques et performances bioclimatiques

Sur le site de votre école XXX sur la parcelle XXX, il vous est demandé de concevoir un pavillon visant à accueillir une nouvelle salle à manger de XX m<sup>2</sup> pour la cafétéria de l'école (cuisine, caisses et sanitaires étant existants, vous n'avez pas à les concevoir). Ce bâtiment doit permettre un confort d'été ET un confort d'hiver aux étudiants de l'école.

Pour cela, vous concevrez une architecture dynamique adaptable en fonction des conditions climatiques extérieures pour atteindre les performances énergétiques visées.

Productions attendues pour le jury final :

- maquette physique dynamique rendant compte de votre projet (maquette globale ou de détail, du 1/10<sup>e</sup> au 1/100<sup>e</sup>) ;

-maquette numérique paramétrique ;

-représentations en 2 dimensions rendant compte de votre projet (plans, coupes et élévations si elles sont significatives).

Cet enseignement vise à permettre à l'apprenant d'approfondir ses connaissances en modélisation paramétrique et de les mettre en œuvre dans le cadre d'un projet de conception architecturale complexe pour lequel les performances énergétiques sont un enjeu majeur.

**- « Le corps et le numérique : une salle de théâtre aujourd'hui » Architecture et web (Atelier Grasshopper et Mosquitos)**

Un enseignement de la modélisation paramétrique en conception architecturale par projet pourrait avoir comme contenu l'enseignement de Grasshopper et du plugin Mosquitos en vue de développer des projets d'architecture prenant en compte l'activité d'utilisateurs sur le web. Le but de ce projet est d'amorcer des réflexions sur l'architecture participative et sur les liens possibles entre des activités sur internet et la conception architecturale.

Pour cela, le projet d'architecture et sa traduction au travers de modèles paramétriques font appel à des savoir-faire techniques liés à : -la conception de modèles paramétriques ; -l'identification sur le web de flux de données pertinents et leur récupération. Le projet fait également appel à des savoirs propres aux sociabilités numériques et aux enjeux de l'implication des utilisateurs dans la conception de l'architecture. Des compétences spécifiques liées au recul critique sur –les données récupérables via internet ; -l'interprétation que l'on peut en donner et ; -leur traduction et leur utilisation en termes géométriques.

Dans le cadre de cet enseignement, nous proposons l'utilisation de Grasshopper pour le développement de modèles paramétriques et le plugin Mosquitos pour récupérer des données de réseaux sociaux et les intégrer aux modèles paramétriques.

Sujet du projet « Le corps et le numérique : un théâtre aujourd'hui »

Architecture et web

Un théâtre veut proposer des résidences pour les troupes développant des pratiques interrogeant les relations de l'écriture au numérique et, interrogeant le corps perçu dans les réseaux sociaux. Il vous est demandé de concevoir un lieu pour les représentations de ces troupes.

Pour cela vous interrogerez l'activité numérique ou le réseau social de votre choix en vue d'en nourrir la définition du programme de cet espace de spectacle et sa conception.

Productions attendues pour le jury final :

- texte argumenté rendant compte des choix pris,
- maquette physique dynamique rendant compte de votre projet (maquette globale ou de détail, du 1/10<sup>e</sup> au 1/100<sup>e</sup>) ;
- maquette numérique paramétrique ;
- représentations en 2 dimensions rendant compte de votre projet (plans, coupes, élévations, schémas).

Cet enseignement vise à permettre à l'apprenant d'approfondir ses connaissances en modélisation paramétrique et de les mettre en œuvre dans le cadre d'un projet de conception architecturale complexe, qui participe d'une réflexion plus large sur le rapport du corps au numérique et de l'intervention des activités de l'utilisateur (ici une activité de discours en ligne) sur le projet architectural.

**- « Passerelle légère » Architecture, structure et fabrication (Atelier Grasshopper, Karamba et Galapagos)**

Un enseignement de la modélisation paramétrique en conception architecturale par projet pourrait avoir comme contenu l'enseignement de Grasshopper et du plugin Karamba en vue de développer des projets d'architecture où la structure et la mise en œuvre du projet lors du chantier seraient tout particulièrement interrogées.

Pour cela, le projet d'architecture et sa traduction au travers de modèles paramétriques font appel à des savoirs techniques liés à l'évaluation des comportements d'une structure et à la prise en compte de ces comportements pour optimiser une architecture. Le plugin Karamba permet d'effectuer des évaluations structurelles rapides de géométries générées sur Grasshopper. Par ailleurs, la fonction « galapagos » de Grasshopper permet d'optimiser une géométrie pour atteindre des performances énoncées.

Sujet du projet « Passerelle légère »

Architecture, structure et fabrication

Il vous est demandé de concevoir une passerelle piétonne sur le site de XX entre XX et XX. Cette passerelle légère devra être structurellement viable et devra pouvoir être mise en œuvre à partir d'éléments préfabriqués pouvant être amenés sur chantier par camion : les éléments préfabriqués ne doivent donc pas dépasser 3 m x 3 m x 6 m.

Vous contrôlerez les performances structurelles de la passerelle grâce à « Karamba ».

Productions attendues pour le jury final :

- texte argumenté rendant compte du fonctionnement structurel de la passerelle et de son système de mise en œuvre ;

- maquette physique dynamique rendant compte de votre projet (maquette globale ou de détail, du 1/100<sup>e</sup> et au 1/10<sup>e</sup>) ;

-maquette numérique paramétrique ;

-représentations en 2 dimensions rendant compte de votre projet (plans, coupes, élévations, schémas).

### **8.3.3.4 Mise en œuvre de l'enseignement**

#### **- Supports didactiques**

Comme pour la pédagogie par travaux dirigés exposée précédemment, dans le cadre de la pédagogie par projet proposée les supports didactiques [dnarchi] et [parametric-ressources] sont utilisés comme des ressources de savoirs et savoir-faire.

Ces ressources sont complétées par un « blog de l'enseignement » explicitement lié à l'institution dans lequel il s'inscrit. Ce blog vise à diffuser : \_la description de l'enseignement (noms des enseignants et des étudiants, dates de rendus, etc.) ; \_ des renvois vers des ressources ou des références abordées en cours ; \_les travaux des étudiants. Ce blog d'enseignement vise à valoriser la production des étudiants.

#### **- Evaluation de l'enseignement**

L'enseignement est évalué lors des deux jurys : le jury intermédiaire et le jury final. Le jury final est composé de personnalités extérieures pertinentes quand au contenu de l'enseignement. L'évaluation porte sur l'adéquation de la production des groupes avec les objectifs exposés en début de cours, sur l'adéquation entre les intentions de projets et les productions des étudiants et-sur les progrès effectués entre le jury intermédiaire et le jury final.

L'évaluation des productions des apprenants permet d'évaluer leur apprentissage et donc la pédagogie mise en place. Celle-ci doit évoluer et être adaptée en fonction des résultats de ces évaluations.



## 9 Conclusion

Cette thèse s'est attachée à construire une connaissance des opérations cognitives en jeu lors d'usages de la modélisation paramétrique en conception architecturale. Le but était de caractériser ces opérations et leurs relations ainsi que d'appliquer les connaissances produites pour proposer une pédagogie de la modélisation paramétrique en conception architecturale.

Dans ce travail, nous nous sommes intéressés en particulier à l'usage des modeleurs paramétriques Grasshopper, Generative Components et Digital Project. Ces modeleurs relèvent d'un mécanisme informatique « basé sur la propagation » qui est un cas particulier de l'approche paramétrique fonctionnelle. Ces mécanismes permettent à un concepteur de définir des chaînes de dépendances entre les éléments constituant son modèle. Lorsqu'il conçoit un modèle paramétrique, un concepteur explicite et contrôle les géométries qu'il définit. Nous avons vu que si ces mécanismes sont puissants, ils demandent de la part du concepteur des compétences spécifiques.

On retrouve l'observation de cette difficulté d'usage des modeleurs paramétriques dans l'état des pratiques proposé dans cette thèse. En effet, il s'avère que l'usage de la modélisation paramétrique en conception architecturale reste la plupart du temps une pratique experte menée par des acteurs différents de ceux de la conception architecturale. Ces pratiques expertes sont menées sur le mode de la consultance externe quand des équipes de spécialistes extérieurs aux agences sont sollicitées pour des missions précises. Et elles opèrent sur le mode de la consultance interne quand les experts ou les équipes d'experts sont intégrés à l'agence. L'état des pratiques mené pointe également les intérêts que les agences trouvent dans le recours à des modèles paramétriques, ces intérêts sont en particulier propres à des missions de fabrication, de communication entre acteurs et de représentation. Cet état des pratiques a amené à interroger les liens cognitifs entre la conception de modèles paramétriques et la conception architecturale.

Nous avons également proposé dans cette thèse un état des aides à la modélisation paramétrique existantes. En particulier, nous nous sommes attachés à l'aide par patterns proposée par Robert Woodbury à partir du concept construit par Christopher Alexander. Nous nous sommes également attachés aux aides publiées sur internet : catalogues de modèles, forums, tutoriels, etc. Ces aides se sont révélées nombreuses. L'usage des patterns et des catalogues de modèles en ligne pour la conception architecturale a été interrogé dans notre recherche du point de vue cognitif.

### **Méthodes mises en œuvre**

Pour caractériser les opérations cognitives en œuvre dans la modélisation paramétrique en conception architecturale, nous nous sommes appuyés sur l'architecturologie. L'architecturologie est un champ qui interroge la conception en termes d'opérations d'attribution de mesures. L'appareillage théorique de l'architecturologie nous a permis d'interroger un corpus de cas d'usages de la modélisation paramétrique en conception architecturale. La méthode d'analyse utilisée relève de l'architecturologie appliquée. Cette méthode propose de lire des indices d'opérations de conception à partir de traces du processus de conception (ces traces peuvent subsister dans des discours, des dessins, maquettes, etc.).

Le corpus analysé dans cette thèse était constitué de cas issus de pratiques professionnelles (des agences Foster and Partners et Gehry technologies entre autres) et de pratiques pédagogiques (issues de diverses écoles d'architecture : ENSA Paris Malaquais, Universitat die Angewandte de Vienne, etc.). Les données ont majoritairement été recueillies par des entretiens, mais nous avons également menés des observations participantes ainsi que des expérimentations. Les données rassemblées sont de diverses natures : ce sont principalement des discours (les entretiens) et des modèles paramétriques, mais il y a aussi des documents graphiques (esquisses de projet, plans, etc.) et des maquettes physiques.

### **Caractérisation des opérations cognitives de conception**

L'analyse architecturologique de notre corpus nous a permis de caractériser les opérations cognitives impliquées lors d'usages de la modélisation paramétrique en conception architecturale. Cette analyse s'est appuyée sur la distinction théorique de deux *espaces de conception* (un *espace de la conception architecturale* et un *espace de la conception de modèles paramétriques*) permettant de distinguer l'activité de modélisation paramétrique comme une activité de conception à part entière. En premier lieu, nos analyses architecturologiques ont pu montrer que les opérations élémentaires de la conception proposées par l'architecturologie (le *découpage*, le *dimensionnement* et la *référenciation*) permettent d'interroger et de décrire des opérations de conception de modèles paramétriques.

Cette première description procède d'une distinction entre opérations de la conception architecturale et opérations de conception de modèles paramétriques. Cette distinction a permis d'interroger finement, par la suite, les relations entretenues entre ces entités. Ces relations peuvent être temporelles : une opération est alors suivie par une autre. Mais ces relations peuvent également être de l'ordre de l'entremêlement par le partage de *pertinence* ou de référence. Ces partages de *pertinences* et/ou de références procèdent d'opérations tierces. Ces opérations tierces sont des opérations de collaboration comme les *opérations de mise en commun* et d'*interprétation* mais aussi des *opérations d'appropriation de l'outil* ou encore des opérations spécifiques à l'usage de modeleurs basés sur la propagation : les *opérations de traduction en géométrie paramétrique*. Ces opérations permettent un entremêlement de la conception architecturale et de la modélisation paramétrique. L'analyse a également montré que ces opérations et leurs entremêlements convoquaient des savoirs spécifiques interrogés sous la forme d'un *espace des savoirs de la modélisation paramétrique*.

L'usage des patterns comme aide à la conception de modèle paramétrique en conception architecturale a également été interrogé d'un point de vue architecturologique. Des expérimentations ont été menées en vue de cette analyse. La recherche a montré que la méthode des patterns est en effet une aide à la conception de modèles paramétriques en conception architecturale mais que celle-ci, pour être effective, nécessite d'être médiatisée par des applications dans des cas concrets : les samples.

### **Définition d'une pédagogie**

Nos résultats nous ont permis de proposer une pédagogie spécifique à l'apprentissage de la modélisation paramétrique en conception architecturale. Les objectifs pédagogiques définis sont liés à la construction par l'apprenant d'un *espace des savoirs de la modélisation paramétrique* mais également aux savoir-faire et compétences nécessaires à la mise en œuvre des opérations cognitives spécifiques à la conception de modèles paramétriques en conception architecturale. Le contexte professionnel et technique de l'usage de la modélisation paramétrique ont permis de définir les objectifs suivant : l'enseignement doit viser une autonomisation de l'apprenant, il doit lui fournir des

ressources qu'il pourra utiliser seul par la suite, l'enseignement doit être centré sur le processus d'apprentissage de l'apprenant.

Dans cette thèse, une pédagogie par travaux dirigés et une pédagogie par projets sont proposées pour répondre aux objectifs pédagogiques et aux objectifs d'enseignement définis. Deux supports didactiques sont proposés en appui de ces pédagogies :

- [dnarchi.fr] une plateforme de publication en ligne d'articles scientifiques vulgarisés qui vise à aider l'apprenant à se construire un *espace des savoirs de la modélisation paramétrique* en particulier et de l'architecture numérique en général ;
- [parametric-ressources] : une plateforme de publication d'une base de données de samples pour assister l'usage de la méthode des patterns.

### **Limites et perspectives de la recherche**

Cette recherche est restreinte à l'analyse de l'usage de la modélisation paramétrique en conception architecturale sous l'angle des opérations de la conception, des opérations pragmatiques et des relations d'opérations. Or, d'autres questionnements architecturologiques, en particulier sur l'*embrayage* et la géométrie, peuvent être soulevés en lien avec ces usages. Par ailleurs, l'opération de *traduction en géométrie paramétrique* proposée dans cette recherche pourrait être plus particulièrement interrogée pour permettre une caractérisation de spécificités cognitives propres aux différentes interfaces des modeleurs paramétriques. Ces trois questionnements architecturologiques sont autant de perspectives possibles pour approfondir les connaissances produites par cette recherche sur les opérations cognitives impliquées dans la conception de modèles paramétriques en conception architecturale.

Par ailleurs, lors de nos analyses, nous avons soulevés le fait que la conception de modèles paramétriques procédait aussi bien de l'utilisation de modeleurs que de l'utilisation de dessins sur supports papier. Ainsi, lors des expérimentations, les étudiants ont tous éprouvés le besoin d'esquisser des intentions de modèles paramétriques sur papier. Si nous reprenons ici la distinction entre *espace de conception* et *espace vrai* posée par l'architecturologie, nous pouvons dire que la conception de modèles paramétriques (qui a lieu dans l'*espace de conception*) semble pouvoir procéder de supports différents (qui relèvent de l'*espace vrai*). Quelles sont alors les rôles de ces différents supports de la conception ? Cette question n'a été que peu abordée dans cette recherche qui, elle, s'est attachée à l'usage des modeleurs sous l'angle de la conception (c'est-à-dire à l'*espace de conception* plutôt qu'à l'*espace vrai*). Qu'est-ce qui préside à l'usage d'un médium plutôt qu'un autre ? Quels en sont les rôles ? De quelles activités cognitives ces choix rendent-ils compte ? sont des questions qui pourront être développées à la suite de cette thèse grâce aux connaissances construites sur les opérations cognitives de conception impliquées.

Sur un autre plan, les ressources didactiques proposées dans le cadre de cette thèse peuvent s'inscrire dans une perspective de développement. Le support didactique [paramétrique-ressources] propose en ligne la diffusion de samples visant à supporter l'apprentissage de la modélisation paramétrique par l'exemple. Actuellement, sur le site [paramétrique-ressources] l'apprenant peut explorer les samples qui l'intéressent en explorant des catégories de samples et en s'aidant des mots clefs définissant les samples. Un moteur de recherche spécifique aux informations contenues par les samples pourrait être pertinent pour la recherche de samples en fonction de requêtes propres à la



conception architecturale ou à la modélisation paramétrique. Pour cela, le développement d'une ontologie paraît pertinent. Les ontologies sont des représentations informatiques visant à rendre compte des connaissances propres à un champ. Cette organisation de la connaissance permet alors de situer des éléments les uns par rapports aux autres, de faire des rapprochements ou des identifications d'objets cherchés. Une ontologie basée sur les connaissances produites dans cette thèse sur les samples (connaissances techniques comme connaissances des activités cognitives impliquées) permettrait d'organiser les samples entre eux et de répondre pertinemment à des requêtes. Cette dernière perspective montre une nouvelle possibilité d'application des connaissances construites par cette thèse.

## 10 Bibliographie

- Agbodan, D., 2002. Nomination persistante dans un modèle paramétrique. Indentification non-ambigüe et appariement générique d'entités topologiques. Thèse de Doctorat, Poitiers: Université de Poitiers, Faculté des Sciences Fondamentales et Appliquées.
- Aish, F. & Chronis, A., 2012. Interview avec Francis Aish et Angelos Chronis (ARD Team, Foster and Partners). (non publié, retranscription insérée en annexe).
- Aish, F. & Joyce, S., 2012. Human Intuition, Computational Rigour. Communication lors du séminaire THP, ENSA Paris Malaquais.
- Aish, R., 2011. DesignScript: origins, explanation, illustration. In Computational Design Modeling. Design Modelling Symposium Berlin. Berlin: Springer.
- Aish, R., 2012. Formation à DesignScript, Workshop AAG. (non publié)
- Aish, R. & Woodbury, R., 2005. Multi-level Interaction in Parametric Design. In A. Butz et al., éd. Smart Graphics. Lecture Notes in Computer Science. Springer Berlin / Heidelberg. Available at: <http://www.springerlink.com/content/2tqyph7pcndkejmq/abstract/> [Consulté le avril 16, 2012].
- Alexander, C., 1966. A city is not a tree. Design, London : Council of Industrial Design.
- Alexander, C., 1977. A Pattern Language: Towns, Buildings, Construction, OUP USA editor.
- Alexander, C., 1971. De la synthèse de la forme, Paris : essai Dunod.
- Alexander, C., 1999. The Origins of Pattern Theory: The Future of the Theory, and the Generation of a Living World. IEEE Software, p.71-82.
- Altet, M., 2006. Les pédagogies de l'apprentissage 2e édition, Paris : Presses Universitaires de France.
- Arnold, M., 1977. Les modèle chez Alexander, Approche critique du Pattern Language, Paris : Centre d'études et de recherches architecturales.
- Bachelard, G., 1934. La formation de l'esprit scientifique 2000<sup>e</sup> éd., Paris : Librairie J Vrin.
- Balmond, C., 2007. Informal, Munich : Prestel.
- Barthelemy, J.-H., 2008. Simondon ou l'encyclopédisme génétique, Paris : Presses Universitaires de France.
- Blanchet, A. & Gotman, A., 2010. L'entretien: L'enquête et ses méthodes 2e édition, Paris : Armand Colin.
- de Boissieu, A. & Guéna, F., 2012. Grasshopper et la programmation sur Rhinoceros 4: une introduction. <http://dnarchi.fr>. Available at: <http://dnarchi.fr/outils/grasshopper-et-la-programmation-sur-rhinoceros-4-une-introduction/>. [Consulté le avril 12, 2012].

- de Boissieu, A., Guéna, F. & Lecourtois, C., 2010a. Enseigner la conception architecturale avec la modélisation paramétrique : Quelle spécificité cognitive ? In Conception assistée par concepteur. 01 Design. Paris: Europaia.
- de Boissieu, A., Guéna, F. & Lecourtois, C., 2010b. Modélisation paramétrique partagée, Le cas de l'utilisation de Digital Project lors de la conception du Pavillon de la Fondation Louis Vuitton pour la Création (Gehry Partners) sous l'angle des opérations de découpage. In Espaces collaboratifs. Scan'1. Marseille.
- de Boissieu, A., Guéna, F. & Lecourtois, C., 2011. « Operation of parametric modelling » and/or « operation of architectural conception » ? Expressing relationships in parametric modelling. In Respecting Fragile Places. eCAADe. Ljubljana.
- Bonora, V., 2012. Interview avec Valerio Bonora (Decode). (non publié)
- Boudon, P., 2004. Conception, Paris: Editions de la Villette.
- Boudon, P., 1991. De l'Architecture à l'épistémologie, la question de l'échelle, Paris: Presses Universitaires de France.
- Boudon, P., 2002. Echelle, Paris: Economica.
- Boudon, P. et al., 2000. Enseigner la conception architecturale : Cours d'architecturologie 2e éd. rev. et augm., Paris: Editions de la Villette.
- Boudon, P., 1971. Sur l'espace architectural Nouv. éd. rev. et augm., Marseille :Parenthèses.
- Boudon, P., Engrand, G. & Lecourtois, C., 2005. Vers un dictionnaire d'architecturologie, Rapport Final, Ecole Nationale Supérieure de Paris la Villette: LAREA.
- Bourbonnais, S., 2010. Les processus expérimentaux : la position décentrée du concepteur. In Conception assistée par concepteur. 01 Design. Paris: Europaia.
- Bourdieu, P., 2003. L'objectivation participante. Actes de la recherche en sciences sociales.
- Burry, J. & Burry, M., 2006. Gaudi and CAD. Journal of Information Technology in construction. Available at: [itcon.org/2006/32/](http://itcon.org/2006/32/).
- Burry, J. & Burry, M., 2012. The New Mathematics of Architecture Reprint., London :Thames & Hudson Ltd.
- Cache, B., 2010. Interview avec Bernard Cache (Objectile). (non publié).
- Cache, B., 1997. Terre meuble, Paris : Hyx.
- Camus, C., 1999. Dire le faire, Présentation d'architectes ou présentation d'oeuvres? , (Les cahiers de la recherche architecturale et urbaine 2-3).
- Camus, C., 1996. Lecture sociologique de l'architecture décrite: Comment bâtir avec des mots?, Paris: L'Harmattan.
- Cannaert, C., 2011. Projective Modelling, Communication pour le Séminaire doctoral de Sint Lukas, Ghent.
- Carmo, M., 2012. The Digital Turn in Architecture 1990-2010 2<sup>e</sup> éd., London : John Wiley & Sons Inc.

- Casilli, A.A., 2010. Les liaisons numériques : Vers une nouvelle sociabilité ?, Paris : Seuil.
- Chadoin, O., 2006. Etre Architecte : Les Vertus de l'Indétermination : De la sociologie d'une profession à la sociologie du travail professionnel, Presses Universitaires de Limoges et du Limousin, Limoges.
- Champy, P. et al., 1994. Dictionnaire encyclopédique de la formation et de l'éducation, Paris: Nathan Université.
- Chien, S.-F. & Yeh, Y.-T., 2012. On creativity and parametric design : A preliminary study of designer's behaviour when employing parametric design tools. In Digital physicality. eCAADe. Prague.
- Cingolani, F., 2012. Network Thinking et apprentissage social: vers une architecture en réseau 2/2. DNArchi. Available at: <http://dnarchi.fr/pratiques/network-thinking-et-apprentissage-social-vers-une-architecture-en-reseau-22/> [Consulté le avril 12, 2012].
- De Coninck, F. & Deroubaix, J.-F., 2012. Transformations des horizons urbains, Savoirs, imaginaires, usages et conflits, Paris: l'oeil d'or.
- Di Cristina, G., 2001. Architecture and Science, London : Wiley-Academy.
- D'Alligna, M., 2009. Interview de Miriam D'Alligna (Foster and Partners). (non publié)
- Davis, D., Burry, J. & Burry, M., 2011. Untangling Parametric Schemata : Enhancing Collaboration through Modular Programming. In Designing Together. CAAD Futures. Liège, Belgique.
- Deleuze, G., 1968. Différence et répétition, Paris: Presses Universitaires de France.
- Deleuze, G., 1988. Le pli - Leibniz et le baroque., Paris: Editions de Minuit.
- Dutant, J., 2010. Qu'est-ce que la connaissance?, Paris: Librairie Philosophique Vrin.
- Dutton, H. & Cingolani, F., 2011. Interview avec Hugh Dutton et Francesco Cingolani (HDA). (non publié)
- Eastman, C. et al., 2008. BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors 2nd Revised edition., London :John Wiley & Sons Ltd.
- Ehret, P., 2009. Interview de Paul Ehret (Gehry Technologies Europ). (non publié)
- Engel, P. & Dutant, J., 2005. Philosophie de la connaissance : Croyance, connaissance, justification, Paris : Librairie Philosophique Vrin.
- Estevez, D., 2001. Dessin d'architecture et infographie : L'évolution contemporaine des pratiques graphiques, Paris : CNRS Editions.
- Everaert-Desmedt, N., 1995. Le processus interprétatif: Introduction à la sémiotique de Ch.S. Peirce, Paris : Mardaga.
- Foucault, M., 1971. L'ordre du discours, Paris : Editions Flammarion.
- Freiberger, M., 2007. Perfect buildings: the maths of modern architecture. plus.maths.org, (42). Available at: <http://plus.maths.org/content/perfect-buildings-maths-modern-architecture> [Consulté le avril 18, 2012].

- Gamma, E. et al., 1993. Design patterns: Abstraction and reuse of object-oriented design. In ECOOP'93 - Object-Oriented Programming, 7th European Conference. Kaiserslautern: Springer, p. 406-431.
- Gehry, F., 2002. 60 II : Frank Gehry. Available at: <http://www.worldcat.org/title/60-ii-frank-gehry/oclc/424361304>. [Consulté le avril 18, 2012].
- Gehry, F., 2007. Interview de Frank Gerhy à l'occasion de l'exposition « Digital Project, Frank Gehry's Vision », Danish Architecture Center. Available at: <http://www.arcspace.com/architects/gehry/dp/dp.html>. [Consulté le avril 18, 2012].
- Girard, C., 1995. Architecture et concepts nomades, Paris : Mardaga.
- Halin, G. & De Luca, L., 2012. Le MAP, un laboratoire pluridisciplinaire et multi-sites. Available at: <http://dnarchi.fr/pedagogie/le-map-un-laboratoire-pluridisciplinaire-et-multi-sites/> [Consulté le avril 18, 2012].
- Harding, J. et al., 2012. Thinking Topologically at Early Stage Parametric Design. In Advances in Architectural Geometry. Paris: Springer.
- Hoffman, C.M. & Joan-Arinyo, R., 2005. A Brief on Constraint Solving, (Computer-Aided Design and Applications).
- Imbert, F. et al., 2012. Concurrent geometric, Structural and Environmental Design: Louvre Abu Dhabi. In Advances in Architectural Geometry 2012. Advanced Architectural Geometry 2012. Paris: Springer.
- Kalay, Y.E., 2004. Architecture's New Media: Principles, Theories, and Methods of Computer-Aided Design, Cambridge : The MIT Press.
- Kocaturk, T. & Medjdoub, B., 2011. Distributed Intelligence in Design, London :Wiley-Blackwell.
- Kolarevic, B., 2005. Architecture in the Digital Age: Design And Manufacturing New Ed., London : Taylor & Francis Ltd.
- Kubota, T., 2012. Influence sur l'architecture de l'évolution de l'outil informatique aux Ateliers Jean Nouvel, Mémoire de Master sous la direction de Guéna, F. et Lecourtois, C., Paris : ENSA Paris la Villette.
- Kwinter, S., 2004. Who's Afraid of Formalism? In Phylogenesis Foa's Ark: Foreign Office Architects. Barcelona: Actar.
- Landa, M. de & Spuybroek, L., 2005. Nox, London : Thames & Hudson Ltd.
- Lecourtois, C., 2005. Architecturologie appliquée à une sémiotique de l'esquisse architecturale. In SCAN05, Rôle de l'esquisse architecturale dans le monde numérique. Paris.
- Lecourtois, C., 2010. Compléxité architecturale et assistance informatique. In DRS 2010. Montreal.
- Lecourtois, C., 2006a. Conception de l'espace et espace de conception. in TIGR (Travaux de l'Institut de Géographie de Reims), les nouvelles approches de l'espace dans les sciences de l'homme et de la société.
- Lecourtois, C., 2004. De la communication sur l'espace : Espace conçu et espace perçu de l'architecture et de l'urbanisme. Thèse de doctorat, Paris: Université Paris X, spécialité Aménagement de l'espace, urbanisme.

- Lecourtois, C., 2008. Enseigner la conception architecturale assistée par ordinateur. In Actes du Colloque BASC2008, Troisième conférence sur l'architecture et le développement durable. Stratégies et perspectives. Biskra.
- Lecourtois, C., 2012. Modéliser la complexité de la conception architecturale numérique, Architecturologie et modèles complexes. In Compléxité(s) des modèles de l'architecture numérique. Paris: Presses Universitaires de Nancy.
- Lecourtois, C., 2006b. Recherche fondamentale et/ou pratique architecturale. Une recherche appliquée propre à l'architecture. In Actes du Colloque EURAU2004. Marseille.
- Lecourtois, C., 2011. Studying collaborative design, Epistemology and research methodology. In EuropIA 13 - Design theory. Rome.
- Lesage, D., 2006. Du dessin à la modélisation déclarative. CAD Magazine #133, p.p.60-62.
- Lindsey, B., 2001. Digital Gehry 1<sup>re</sup> éd., Basel : Birkhäuser.
- Liveneau, P. & Marin, P., 2012. Materiality in its contemporary forms, Architecture, perception, fabrication, conception Isles d'Abeau : Les grands Ateliers.
- Lynn, G., 1997. Animate Form Har/Cdr., Princeton Architectural Press.
- Maculet, R. & Daniel, M., 2005. Conception, modélisation géométrique et contraintes en CAO: Une synthèse, Marseille: Laboratoire des Sciences de l'Information et des Systèmes UMR CNRS 6168.
- Malé-Alemany, M., 2009. Parametric Construction an exploration on virtual standardization. In Models. Princeton Architectural Press.
- Marin, P., 2010. Exploration des mécanismes évolutionnaires appliqués à la conception architecturale. Mise en oeuvre d'un algorithme génétique guidé par les qualités solaires passives de l'enveloppe. Thèse de doctorat, Nancy: Institut National Polytechnique de Lorraine.
- Marques, D. & Woodbury, R., 2007. Managing Contingency in Parametric Models through Implicit Relational Modeling. In A. Dong, A. V. Moere, & J. S. Gero, éd. Computer-Aided Architectural Design Futures (CAADFutures) 2007. Springer Netherlands, p. 279-288. Available at: <http://www.springerlink.com/content/v2m248r887h41p10/abstract/> [Consulté le avril 16, 2012].
- Marques, D. & Woodbury, R., 2006. Using Rule Based Selection to Support Change in Parametric CAD Models. In A. Butz et al., éd. Smart Graphics. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, p. 230-35. Available at: <http://www.springerlink.com/content/513t7r6454012x52/abstract/> [Consulté le avril 16, 2012].
- Meirieu, P., 2004. Enseigner, apprendre à l'université. Available at : <http://www.meirieu.com/OUTILSDEFORMATION/> [Consulté le avril 16, 2012]
- Menges, A., 2006. Instrumental geometry. in Architectural Design.
- Migayrou, F., 2003. Architectures non standard, Paris : Editions du Centre Pompidou.
- Mineur, Y., 2011. Interview avec Yves Mineur. (non publié).
- Mineur, Y., 2009. Introduction aux outils numériques d'aide à la conception, Eléments de synthèse du cours 2A-ME-ENIAME. (non publié).

- Monnin, A. & Halpin, H., 2012. Toward a Philosophy of The Web. *Metaphilosophy*, 43(4). Available at: <http://onlinelibrary.wiley.com/doi/10.1111/j.1467-9973.2012.01764.x/abstract> [Consulté le mars 14, 2013].
- Morel, P., 2012. Quelques remarques sur les enjeux pour toute école d'architecture de la «(non)représentation » à l'ère du numérique. Available at: <http://dnarchi.fr/pedagogie/quelques-remarques-sur-les-enjeux-pour-toute-ecole-darchitecture-de-la-nonrepresentation-a-leredu-numerique/> [Consulté le mars 14, 2013].
- Morvan, P., Detollenaere, N. & Meinadier, J.-P., 2000. Dictionnaire de l'informatique , Acteurs, concepts, réseaux Larousse., Paris. Available at: <http://livre.fnac.com/a114730/Pierre-Morvan-Dictionnaire-de-l-informatique> [Consulté le octobre 11, 2012].
- Neil, L., 2002. Swarm Tectonics. *Archis*. Available at: <http://neilleach.wordpress.com/>. [Consulté le mars 14, 2013].
- Nikolinakou, M., Tallon, A. & Ochsendorf, J., 2005. Structure and Form of Early Gothic Flying Buttresses. *Revue Européenne de Génie Civil*, 9.
- Nouvel, J., 2010. Interview de Jean Nouvel pour Imagina 2010. Available at: <http://www.lemoniteur.fr/153-profession/video/696628-entretien-avec-l-architecte-jean-nouvel-au-salon-imagina-2010> [Consulté le mars 14, 2013].
- Occhipinti, E., 2011. Communication lors de la journée « Formes Complexes » 2.0. Paris.
- Pachiaudi, A., 2011. Interview avec Alexandre Pachiaudi (RPBW). (non publié).
- Payne, A. & Issa, R., 2009. Grasshopper Primer for version 0.6.0007, Available at: [\[grasshopper3d.com\]](http://grasshopper3d.com) [Consulté le mars 14, 2013].
- Peev, S., 2012. Interview avec Svetlin Peev. (non publié).
- Picon, A., 2012. Architecture numérique, Conférence introductive de la conférence SCAN12. Paris.
- Picon, A., 2010. Culture numérique et architecture - Une Introduction 1<sup>re</sup> éd., Paris: Birkhauser
- Potamianos, I., Turner, J. & Jabi, W., 1995. Exploring the Proportions of Middle-Byzantine Churches: a Parametric Approach. In *The Global Design Studio: Proceedings of the Sixth International Conference on Computer-Aided Architectural Design Futures. CAAD Futures*. Singapore.
- Pottmann, H. et al., 2007. *Architectural Geometry*, Exton : Bentley Institute Press.
- Qian, C.Z., Chen, V.Y. & Woodbury, R., 2007. Participant Observation can Discover Design Patterns in Parametric Modeling. In *Expanding Bodies. ACADIA*.
- Quintrand, P., 1990. *La Conception assistée par ordinateur en architecture*, Paris : Hermes Sciences Publicat.
- Quivy, R. & Campenhoudt, L.V., 2006. *Manuel de recherche en sciences sociales 3e édition revue et augmentée.*, Paris: Dunod.
- Ben Rajeb, S., 2012. Modélisation de la collaboration distante dans les pratiques de conception architecturale : Caractérisation des opérations cognitives en conception collaborative instrumentée. Thèse de Doctorat, Paris: Ecole d'Architecture de Paris la Villette.
- Rey-Debove, J. & Rey, A., 1992. *Le Petit Robert édition 1992.*, Paris : Le Robert.

- Roussel, M., 2012. De l' « Alien » à l' « Allo- ». Devenir-alien, allogenèse et vie artificielle dans la transarchitecture de Marcos Novak. Available at : <http://dnarchi.fr/culture/de-l-alien-a-l-allo-devenir-alien-allogenese-et-vie-artificielle-dans-la-transarchitecture-de-marcos-novak/> [Consulté le mars 13, 2013].
- Rutten, D., 2010. Evolutionary Principles applied to Problem Solving. <http://www.grasshopper3d.com>. Available at: <http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles> [Consulté le mars 13, 2013].
- Schiftner, A. et al., 2012. Architectural Geometry from Research to Practice: The Eiffel Tower Pavilions. In *Advances in Architectural Geometry 2012. Advanced Architectural Geometry 2012*. Paris: Springer.
- Schumacher, P., 2008. Parametricism as Style, Parametricist Manifesto. Available at: <http://www.patrikschumacher.com/Texts/Parametricism%20as%20Style.htm>. [Consulté le mars 13, 2013].
- Shelden, D., 2002. Digital Surface Representation and the Constructibility of Gehry's Architecture. Thèse de Doctorat, Massachusetts Institute of Technology.
- Shelden, D., 2006. Interview with Dennis Shelden, Chief Information Officer, Gehry Technologies. Available at: <http://www.zdlaw.com/enews/2006-vol11-num4-01.php>. [Consulté le mars 13, 2013].
- Simondon, G., 1958. *Du mode d'existence des objets techniques*, Paris : Editions Aubier.
- Smith, R., 2005. *Changing the shape of architecture digitally*, (non publié).
- Smith, R., 2011. Interview avec Rick Smith (Gehry Technologies). (non publié).
- Soulé, B., 2007. Observation participante ou participation observante? Usages et justifications de la notion de participation observante en sciences sociales. *Recherches Qualitatives* n°27, p.124-140.
- Steele, J., 2002. *Architecture and Computers: Action and Reaction in the Digital Design Revolution* 1st éd., Watson-Guptill Publications.
- Stiegler, B., Meirieu, P. & Kambouchner, D., 2012. *L'école, le numérique et la société qui vient*, Paris : Fayard/Mille et une nuits.
- Sutherland, I.E., 1963. *Sketchpad : A man-machine graphical communication system*. Thèse de Doctorat, Cambridge: University of Cambridge.
- Szalapaj, P., 2000. *CAD Principles for Architectural Design: Analytical Approaches to Computational Representation of Architectural Form*, Oxford : Architectural Press.
- Tapie, G., 2000. *Les Architectes : mutations d'une profession*, Paris : L'Harmattan.
- Terrin, J.-J., 2005. *Maîtres d'ouvrage, maîtres d'oeuvre et entreprises : De nouveaux enjeux pour les pratiques de projet*, Paris : Eyrolles.
- De Terssac, G. & Chabaud, C., 1990. Référentiel opératif commun et fiabilité. In *Les facteurs humains de la fiabilité dans les systèmes complexes*. Toulouse.
- Terzidis, K., 2006. *Algorithmic Architecture* 1<sup>re</sup> éd., Oxford : Architectural Press.



- Tiazzoldi, C., 2012. Le modèle comme acte créatif et innovation combinatoire, Le projet Onion Pinch. dnarchi.fr. Available at : <http://dnarchi.fr/pratiques/le-modele-comme-acte-creatif-et-innovation-combinatoire-le-projet-onion-pinch/> [Consulté le juillet 5, 2012].
- Triclot, M., 2011. Philosophie des jeux vidéos, Paris : editions Zones.
- Vignal, M., 2010. Frank O. Gehry: « Je cherche à humaniser la modernité ». L'EXPRESS. Available at: [http://www.lexpress.fr/culture/art/frank-o-gehry-je-cherche-a-humaniser-la-modernite\\_849534.html](http://www.lexpress.fr/culture/art/frank-o-gehry-je-cherche-a-humaniser-la-modernite_849534.html) [Consulté le juillet 5, 2012].
- Violeau, J.-L., 2005. Les architectes et Mai 68, Paris : Recherches.
- Violeau, J.-L., 1999. Quel enseignement pour l'architecture ? Paris : Editions recherche, école d'architecture Paris-Belleville.
- Weisberg, D.E., 2008. The Engineering Design Revolution; The People, Companies and Computer Systems That Changed Forever the Practice of Engineering, cadhistory.net. Available at: [cadhistory.net/](http://cadhistory.net/) [Consulté le juillet 5, 2012].
- Whitehead, H., 2009. Interview avec Hugh Whitehead (SMG, Foster and Partners). (non publié, retranscription insérée en annexe)
- Witt, A., 2008. Entretien avec Andrew Witt, GT (ArchiCrée n339). Paris.
- Witt, A., 2012. Euclid :An Open, Cross-Platform, Cloud Geometry Optimizer and Library. In Advances in Architectural Geometry 2012. Paris: Springer.
- Witt, A., Nolte, T. & Shelden, D., 2009. Le design concomitant à grande échelle: le cas de la fondation Louis Vuitton. Communication non publiée.
- Woodbury, R., 2010. Elements of Parametric Design, London: Routledge.
- Woodbury, R., Aish, R. & Kilian, A., 2007. Some Patterns for Parametric Modeling. In Expanding Bodies. ACADIA. Halifax (Nova Scotia), p. 222-229. Available at: [http://cumincad.scix.net/cgi-bin/works/Show?\\_id=acadia07\\_222&sort=DEFAULT&search=robert%20woodbury%202007&hits=742](http://cumincad.scix.net/cgi-bin/works/Show?_id=acadia07_222&sort=DEFAULT&search=robert%20woodbury%202007&hits=742) [Consulté le avril 16, 2012].

## Sites Webs cités

|                                |   |
|--------------------------------|---|
| [architecturables.org]         | <a href="http://architecturables.org/">http://architecturables.org/</a>   |
| [arcspace.com]                 | <a href="http://www.arcspace.com/exhibitions/unordered/digital-project/">http://www.arcspace.com/exhibitions/unordered/digital-project/</a>                               |
| [bentley.com/gc]               | <a href="http://www.bentley.com/gc">http://www.bentley.com/gc</a>   |
| [cnrtl.fr/definition/]         | <a href="http://www.cnrtl.fr/definition/">http://www.cnrtl.fr/definition/</a>   |
| [co-de-it]                     | <a href="http://www.co-de-it.com/">http://www.co-de-it.com/</a>   |
| [codelab]                      | <a href="http://codelab.fr/">http://codelab.fr/</a>   |
| [codequotidien]                | <a href="http://codequotidien.wordpress.com/">http://codequotidien.wordpress.com/</a>   |
| [Complexitys.com]              | <a href="http://complexitys.com/">http://complexitys.com/</a>   |
| [culturecommunication.gouv.fr] | <a href="http://www.culturecommunication.gouv.fr/Disciplines-et-secteurs/Architecture/">http://www.culturecommunication.gouv.fr/Disciplines-et-secteurs/Architecture/</a> |
| [decodebim.com]                | <a href="http://www.decodebim.com/">http://www.decodebim.com/</a>   |
| [designpatterns.GH]            | <a href="http://www.andrew.cmu.edu/org/tsunghsw-design/">http://www.andrew.cmu.edu/org/tsunghsw-design/</a>   |
| [designpatterns]               | <a href="http://www.designpatterns.ca/">http://www.designpatterns.ca/</a>   |
| [designreform.net]             | <a href="http://www.designreform.net/">http://www.designreform.net/</a>   |
| [Designtoproduction.com]       | <a href="http://www.designtoproduction.com/">http://www.designtoproduction.com/</a>   |
| [digital-knowledge.net]        | <a href="http://www.digital-knowledge.net/">http://www.digital-knowledge.net/</a>   |
| [digitaltoolbox]               | <a href="http://digitaltoolbox.info/">http://digitaltoolbox.info/</a>   |
| [dnarchi.fr]                   | <a href="http://dnarchi.fr/">http://dnarchi.fr/</a>   |
| [evolute.at]                   | <a href="http://evolute.at/">http://evolute.at/</a>   |
| [fosterandpartners.com]        | <a href="http://www.fosterandpartners.com/">http://www.fosterandpartners.com/</a>   |
| [gehrytechnologies.com]        | <a href="http://gehrytechnologies.com/">http://gehrytechnologies.com/</a>   |
| [GHresources.blogspot.com]     | <a href="http://grasshopperresources.blogspot.fr/">http://grasshopperresources.blogspot.fr/</a>   |
| [giuliopiacentino]             | <a href="http://www.giuliopiacentino.com">http://www.giuliopiacentino.com</a>   |
| [grasshopper3D.com]            | <a href="http://www.grasshopper3d.com/">http://www.grasshopper3d.com/</a>   |

|   |   |
|---|---|
| [gteam.com]   | <a href="http://www.gteam.com/">http://www.gteam.com/</a>   |
| [Gt-wiki]   | <a href="http://www.gtwiki.org/">http://www.gtwiki.org/</a>   |
| [ <a href="http://www.data.gouv.fr/">http://www.data.gouv.fr/</a> ] | <a href="http://www.data.gouv.fr/">http://www.data.gouv.fr/</a>   |
| [jeannouvel.com]  | <a href="http://www.jeannouvel.com/">http://www.jeannouvel.com/</a>   |
| [labs.autodesk.com]   | <a href="http://labs.autodesk.com/utilities/designscript">http://labs.autodesk.com/utilities/designscript</a> |
| [MAP]   | <a href="http://www.map.archi.fr/">http://www.map.archi.fr/</a>   |
| [mathcurve]   | <a href="http://www.mathcurve.com/">http://www.mathcurve.com/</a>   |
| [mc2012]  | <a href="http://mc2012.sciencesconf.org/">http://mc2012.sciencesconf.org/</a>                                 |
| [MorphoCode]  | <a href="http://morphocode.com/">http://morphocode.com/</a>   |
| [nzarchitecture.com]  | <a href="http://www.nzarchitecture.com/blog/">http://www.nzarchitecture.com/blog/</a>                         |
| [parametricmodels]  | <a href="http://www.parametricmodel.com">http://www.parametricmodel.com</a>                                   |
| [parametric-ressources.com]   | <a href="http://www.parametric-ressources.com/">http://www.parametric-ressources.com/</a>                     |
| [ParaMod]   | <a href="http://paramod2.blogspot.fr/">http://paramod2.blogspot.fr/</a>                                       |
| [sanfrancisco.aaschool]   | <a href="http://sanfrancisco.aaschool.ac.uk/">http://sanfrancisco.aaschool.ac.uk/</a>                         |
| [scratch.mit.edu/ ]   | <a href="http://scratch.mit.edu/">http://scratch.mit.edu/</a>   |
| [smartgeometry.org]   | <a href="http://www.smartgeometry.org/">http://www.smartgeometry.org/</a>                                     |
| [Suckerpunch]   | <a href="http://www.suckerpunchdaily.com/">http://www.suckerpunchdaily.com/</a>                               |
| [studiolynn.at]   | <a href="http://www.studiolynn.at/">http://www.studiolynn.at/</a>   |
| [tex-fab.net]   | <a href="http://tex-fab.net/">http://tex-fab.net/</a>   |

# Sommaire des annexes

## **ANNEXE 1 : Retranscription d'entretiens**

|   |  |     |
|---|--|-----|
| 1 | Corpus de données : retranscription d'entretiens ..... | 327 |
|---|--|-----|

## **ANNEXE 2 : Analyses architecturologiques**

|   |  |     |
|---|--|-----|
| 2 | Analyse de pratiques d'étudiants.....  | 340 |
| 3 | Analyse de pratiques professionnelles .....                                  | 377 |
| 4 | Analyse architecturologique de cas documentés par les expérimentations ..... | 389 |

## **ANNEXE 3 : Catalogue de samples**

|    |  |     |
|----|--|-----|
| 5  | Organiser des arbres de données .....                                  | 417 |
| 6  | Créer des points.....  | 423 |
| 7  | Créer des courbes .....  | 434 |
| 8  | Créer des surfaces.....  | 446 |
| 9  | Créer des motifs .....   | 456 |
| 10 | Evaluer .....  | 463 |
| 11 | Récapitulatifs des samples proposés et de leurs caractéristiques ..... | 468 |

## **ANNEXE 4 : Retours sur enseignements**

|    |  |     |
|----|--|-----|
| 12 | Retour sur les enseignements so823-2011 et so823-2012..... | 475 |
|----|--|-----|

# ANNEXE 1 :

## Retranscriptions d'entretiens

### 1 Corpus de données : retranscription d'entretiens

#### **1.1 *Retranscription de l'interview d'Hugh Whitehead du 29-11-2009 Foster and Partners (specialist modeling group)***

AB // Are the intents and the meaning of the design teams (always) clear? How are the exchanges between the Specialist Modelling Group and the design groups? (Meeting, drawings, texts, schemes, etc.) Are you implicated in the architectural design? or are the intents already well-defined when you are intervene?

HW // In the office, we act like an in-house consultancy. The office has now six design groups; each group is about 150 people. They're all part of one culture but the groups work in competition with each other sometime. Consultancy groups are in the office and there are many: the model shop, visualization group, space planning and SMG is one of those. We all span all over the design groups. To some extent it is a support draw but in all case it is very interpretive. We act as consultants so we need to understand the project and understand the mindset of the design team. When we start to work on a project we like to feel like we are part of the team. Also, we do a lot of work in conjunction with external consultants (structural, environmental, etc). Here is a very interesting triangular relationship. We work very closely with the design team, we work with excellent consultants and we try to bring it all together.

In term of kind of brief of that we get, sometimes a team comes to us early when they think they had a problem, quite often it's too late! There is always something you can do, but we prefer to be involved very early so that we can help to steer the project.

AB// Do you intervene also during the sketch of the project?

HW // Yes, even at that stage. All the projects in the office go to present to the design board review. We often attend to the design board review, so we hear all the comments. The design intent of (the stair?) for all the projects comes through the design board. But there are a lot of responsibilities at all level to be adventurous and take the project to new territories.

AB// Do you propose things sometimes?

HW// Yes, in the early stages of the project. We often call it "option earring" (hearing?). The design team has to produce many alternatives, and then they take the best of those alternatives and present it at the design board. Practically, using parametric modelling or scripting techniques, we can build models which are options generases (?). That's a significant part of our work. We see the potential to build the model which has ability to do fan tuning (?) and optimization.

But there is a trap here. Because very often the design will suddenly completely change direction and when that happens, the parametric modeller, the script, has to take a new configuration. So always we have to take very careful not to invest too much time and effort too early in something that is totally mechanistic.

We make a distinction between process and procedure. In the early stages of the design it is very much in open ended process. We don't even know the starting point or the intended rules or the result. But once you have a configuration which is stable, then you can use procedural approaches. So we try to use procedures which are quite small and modular. And then they can be recombining in a way that accelerates the process. So we look for a very generic and loose fit strategy for this type of work.

AB// These procedures can be small parts of the structure? Parts of the building?

Yes exactly. It could be structural, environmental or whatever. When you have a configuration that is stable then you can begin to develop parameter sense (?) but also option switches. So very often you will do a script way to have the alternative to select different option strategy then produce different type of model. You can only do this if you have a very close dialogue with the design team. And it's important to say, we do not do that work for them, we trying to inform their work and accelerate the process.

AB// Are you working on all the projects?

Well, on too many! We always have a priority stack, but now the team is about ten people and the office is more than a thousand so it's heavy multi-plan. This means that we can develop our effort the most only on challenging projects. Which is very exciting for us but it is also very demanding. So when we do that, sometimes the project is on a short duration, sometimes for a competition or maybe to resolve specific issue. Other times there is a much longer duration, in some way it is better for us. We can do more significant development work during the longer period. We always looking for the opportunity to take the development or customization and refine it in a way that can be use by other projects and other teams. We're always looking for the opportunity to reinvest all the time we spend.

AB// So you are making your own development on your software? Writing your script?

All of the people of the team began as architects, and then develop other skills on top. We think that we are “architects +”. That’s very important for understanding the design intent and playing the interpretative role. But all the people of the team now have quite highly develop scripting ability in several different languages at different depth. We also look for opportunities to combine different types of consultancy. Parts of our group are environmental (?) also we work closely with sustainability researches and also we have an urban design group. They sit on the same bench all of there, on different group (?).

AB// In your paper in the Kolarevic’s book it seems like the members of your team were mostly engineers. So it evolves?

No, we’ve always been mainly architects. One member of our team, Francis Aish (who is the son of Robert Aish) his background is aeronautic engineering. He is the only person of the team who is more engineer than architect. But in ten years he learns a lot about architecture.

AB// Is the philosophy of the team evolves with experiences?

Yes of course. Somebody said that Foster is the culture that reinvents himself everyday, we like to feel that we are part of that. We always do a lot to review where we are and try to evolve the strategy for where we might be next. We now have SMG’s members who are part of design group, but they relay closely to us. Often in the evenings we have a SMG group’s session. We show the later things we’re doing and the SMG’s member show the late things from the group. We try to get cross-fertilization. That’s very important if we want to stay one culture in the office. Competition is a good thing, is healthy, but also you need to find ways to get people to share knowledge.

AB// Can we speak on specific project? The city hall for example, could you explain me the intents, the “story” of the project?

The story of this project began when the partner in charge came by, and that’s just happen to him we are doing something by the river, I think it’s a pebble. It wasn’t an instruction but I decided to look at it. How do you model a pebble shape? At this time there wasn’t very much software which help to do free form design. I try to look at it from first principles. One think I knew was the starting point would be a sphere and the sphere will deform, along axis, and axis points toward the sun. This was a key idea of the design: by doing that you have a small area at the top, which take most of the solar radiation. That area you populate with PV cells to trap energy. Then with the slop of the side, at the back, you need to defend against the solar radiations. The profile building is in steps, the back of the building became self shadowing.

AB// In the first idea it was already like this?

No, it develops through a lot of sketches. It started as a pure form, a sphere that had been deformed.

AB// And what about the site? The context?

It’s by the river, on the bank of the river. And at one time even, it was the idea to bring the river in, to a dock and have the building sitting in water. Because it public building the view it take, it makes it looks defensive: like a castle in a mote. So they said that’s the wrong impression to give, we want to make the building feel accessible to everyone from everywhere. Because the purpose of the

building is city hall, where they want to make politics a public process where people can go and hear debates and understand the political process. The idea was to create transparency for the building has to symbolize that.

AB// The first step was a “pure shape”?

Yes. It’s uh... We started to make this just using NURBS modelling.

AB// On microstation with the DDD (Dimensional Driven Design) ?

Yes. Points here [montrant une illustration extraite de (Kolarevic 2005)] are part of the control polygon, this control polygon create the surface. So first of all we do experiment to decide what the minimum control polygon that you need is. If you make a sphere on Rhino and if you make some cutting operations and so on, if you look at your control polygon it’s very very dense and this make the surface difficult to work with. It keeps smooth but when we turn the surface to a solid, you can have minimum control polygon, so you can make some cut and there is no degradation. So the advantage of uses a control polygon directly you can control tangency, also you can employ proportional relationships, this is how we use the DDD profile. By changing the proportions of these few points of the control polygon, you affect the rate of change of curvature. So this really is the heart of free form design: you are trying to control curvature by rate of change. In mathematics you have your second differential or third if you like. This is really the philosophy behind it. This help to have a very simple mechanism with very few inputs. We try to make a tool we can give to the designers, where they control they understand the mechanism, they maybe don’t understand the details of how it works, but through this we create an interface. We publish to them the driving proportions. Also these proportions, you don’t need to enter a number, you can just drag and stretch: so they understand by using it, intuitively, how to change shape.

In order to make such a mechanism, we have to predict what the designers want to do, what changes they need to make. In this example here [the DDD model for the City Hall in (Kolarevic 2005)] you see that they can stretch it, they can make this part of the curvature titer?? so that this have a camp??? and this have a flatten end. Then also it can be asymmetric about the axes, here this curve is extending down-??? because this is where most of the accommodation happens. But we need to limit the depth because we still want the self shading angle to etc etc. There are many uncertainties to plan with the shape. Also you can see in the plan view there is also flatten. The sphere became more like an egg, but even the egg itself is flatted at the top with the axis inclines. How do you transform a sphere to an egg purely with proportions? If you thing you need terms you can draw the diagram for the control polygon then you start controlling dimensions, but then everything became out of proportion.

The curvature proportions for the end are in fact total ended to the axis, so if the designer changes the length of the axis, the proportions remain. So in this way we are predicting that they may want to extend or reduce the axis, but they already set curvature proportions which they want to retain. This is an example of how we have to predict the changes they need to make.

Once they start to play with it we work with them and we listen to their comments. Usually they say something like “Ah, this is very good, but I would like to also do this”. Then we have to go away and thing how we extend the mechanism so it would allow another transformation or a different control mechanism. We try to get back to the roots of geometry and do everything by proportion and



express that in a mechanism and publish the controllers in a simple interface. Interface is always with sliders so you can intuitively play with it.

AB// Do many models have been done for the project?

Yes many models! (rire) The next stage for this process was to slice the form in horizontal cut. The result of that was to generate the floor plates. And that point the interior designers and the space planners look at the floor plates shapes and then they start to decide what would be the position of the core and how will the layout works floor by floor. So then they start to give feed back to the design team saying “Some of those shapes didn’t work for us, can you change the shape to give us more space?”. We got interaction between all the teams evolved. In the same time the environmental analyse were studies to look at the radiation on the panels.

It was a long time ago now, but it put in place a certain idea of certain principles that in some extend we still follow but we do a lot more in GC, in GH and other applications. We do a lot more custom scripting. In the same time we develop our function libraries. When our team is scripting, they can share some of the code made by other people. The role of Francis has change really doing everything, now is more the bank. Creating the structure for the library and helping promote ideas and so on.

## **1.2 Retranscription de l’interview de Francis Aish et Angelos Chronis, du 15-05-2012**

Francis Aish : Responsable de l’équipe ARD de Foster and Partners

Angelos Chronis : membre de l’équipe ARD de Foster and Partners

Aurélie de Boissieu : Chercheur au lab. MAP-maacc

L’entretien a eu lieu le 15-05-2012 dans les locaux de l’agence Foster and Partners. L’entretien a duré 50 minutes. Des thèmes avaient été envoyés la veille *via* mail par l’intervieweur, Aurélie de Boissieu, le guide d’entretien quant à lui a été soumis à Francis Aish et Angelos Chronis le jour même.

**Francis Aish//** Until one year ago, the ARD and the SMG were the same team. Hugh Whitehead and I were running the SMG together. So it comes from the same background. But about a year ago, we had this discussion within the company about the potentiality to try different approaches and strategies on way of doing things. It was decided to split the SMG. I now run the ARD team (ARD for Applied Research and Development). Hugh is still running the SMG geometry and then Irene Gallou is running the SMG environmental. So it is a three way split. My observation with that is: we have been successful with parametric modelling and with early environmental stuff. But that become more and more mainstream. This is a great thing but I wanted to do more ARD like the name of the team suggests... Another point is that doing pure RD is not relevant. It’s a head to be applied, but we have to produce a work on many project.

**Angelos Chronis//** We have been working on also sixty projects the past year.

**Francis Aish//** Right, sixty-seventy... And the team has moved now to seven people, with the average, maybe five, of the associates. It's quite a lot of projects for a small team. That keeps the research very relevant. There is no option to do very abstract stuff... But we are also doing long-term research.

**Angelos Chronis //** And whatever we try to do we look for re-use.

**Francis Aish//** Half the team is doing a PhD. So we are doing long-term research. But there is a risk: research can be intellectually very interesting, but if it can't be applied... That's maybe fine in academia but in industry not.

**Aurelie de Boissieu//** Does it mean that today the SMG is doing more "mainstream" modelling?

**Francis Aish//** The SMG is doing more modelling, but a lot of research as well. We are still from the same origin... The SMG looks more at new ways of interacting with design tools. That is interesting with modelling and environmental analysis. And the office is about 1200 people, there are still demands there.

**Aurelie de Boissieu//** Few years ago H. Whitehead spoke about an ambition to build an environment to share codes, did you managed with this project know?

**Francis Aish//** We share things at two levels. We wrote a lot of custom tools on specific projects. All our projects have common things but are all quite mainly different. As we are a small team we try to reuse things as much as possible. We haven't got time to rewrite everything from scratch, every single time.

We use things at two levels. We try to generalize our tools and make them available to the entire office. That appears in the main use in Microstation, in Ecotech, and in other packages. That's one way of sharing things. And then internally we share code as well. As you mentioned before we try very modular code and building blocks. We also have a function library that we code all together which we can reuse. We can re-use things programmatically.

There is an interesting bond in sharing source code and having an API and a library. On one hand what is very good in sharing source code, is that you can learn a lot. On the other hand there is a risk when every one can make small changes. Here we have to be pragmatic. Having an API is more efficient. Functions are not always doing the right things and you have to extend it.

The company is big, more and more people are running code: we need to be doing a bit of software engineering. We are in very large light weight fashion, we have to do that.

**Angelos Chronis//** We do share codes between us in database.

**Francis Aish//** Yes, absolutely. It is also easier with a smaller team. The all SMG team is quite large now, 17 persons use it to share things.

**Aur lie de Boissieu //** Are these database shared with everyone here?

**Francis Aish//** No. The code database is kept within the team.

**Angelos Chronis//** Some are shared, not the whole, but applications are shared.

**Francis Aish//** Yes the tools are shared across the office, so anyone can use them. But the source code is kept.

**Aurélie de Boissieu** // Libraries are shared too?

**Francis Aish**// Libraries are available. Anyone can use the library, but not the source code. The people can make contribution to that. Anyone in the office can say “I brush my own tool, and go, you can take this out”.

We need to keep a little bit of control over the process. That seems to work quite well. Because our role is more general and broader, we can spend a little more time in generalizing things and code. People on the Design Groups have less time to make their code general purpose. Sometimes we can take that code and we can reuse it.

**Angelos Chronis**// In the Design Groups there is a lot of people that are scripting in GC, GH... We can always share a GC script that can create geometry. But when we speak about code there are very few people that can deal with that. So essentially you see a lot of people in the Design Team who work with GH and scripts but not with code as we do.

**Francis Aish**// Yes. I separate scripting and programming. When I first started, there were very few people in the office doing programming for the design. Five years ago maybe most of the SMG was doing most of scripting. Now a lot in Design Team can deal these skills. They can use GH and GC. That have been very liberating, because now we can focus in programs more depth. We still do a lot of “scriptfull” projects, but we can go deeper. There is a lot more people that can do the basics now. That is very useful. (rire) Being at the bottom is so much fun. (rire)

**Aurelie de Boissieu** // What about scripting, do you teach it? Do you help design groups with sharing scripts?

**Francis Aish**// In the past we have done some teaching. I have taught some VBA courses about 4 years ago now and we ran GC workshops as well...

**Aurelie de Boissieu** // In-house?

**Francis Aish**// In house. That worked pretty well. We find as well people who picked up GH by themselves or people from course that Angus and Mark are teaching. In a lot of general architecture course now people are exposed to scripting. So we are doing a little less teaching now than we used to. Because we don't need to: the skills are there. Some people have a natural aptitude for it, some people need a little bit of help. But even if they don't write the scripts themselves, if they understand the process that is very helpful because they can specify to you what they really need. If you had never really scripted it will be hard to explain your problem, but if they script they can at least said: “this is the way my problem is structured”.

We write help files, we make tools as accessible as possible. We had a lot of work the last year on trying to make the interfaces tool very accessible. We had a lot of work with i-pad and touch screens. Very often people can interact with these tools in an intuitive way. You can then shorten the decision making process. That is very important, especially nowadays for the clients that are very demanding. Time scales are very short. In a Design Review if a Partner says “I change that, I fix that”, he can see the response... That is very powerful.

**Angelos Chronis//** We are creating a lot of user interfaces, that we try to simplify. Sometimes you get feedbacks that are very complex like drafts, values... We try to simplify that the more as possible for people to understand it really clearly.

**Francis Aish//** There is a lot of depth behind it, but you allow quick decision to be taken, and quick regress decision.

**Aurelie de Boissieu //** On which applications are you currently working on?

**Francis Aish//** On a lot of things: Microstation,... Because we are a small team, if there is an existing tool we will use it, if we can build on the top of, we will use that, if nothing exist we will write it from scratch.

Microstation is the office CAD platform with GC. A lot of people use Rhino and GH as well. We use custom software on processing, C++, Python, VBA (Microstation), VB, Java, ... We use a very big range of tool.

If we can reuse things we will do. Because it is a small team and there are so many projects on the office, we have to be efficient. We try many things as successful as possible. If we write on Microstation we can deploy across the office very quickly... This is not about speeder one team; this is about speeder the entire office. The office moves slower than the small team can do. But a lot of people move faster, that is important.

**Aurelie de Boissieu //** Do you have specific moments to present your new tools to the different Design Teams?

**Francis Aish//** Yes we have sort of office presentations, we have internet... A lot of it is a sort of word of mouth, we meet a lot of people because they walk around the teams, and they tell someone "ok you work on GH, ok I will look about that"...

**Angelos Chronis//** And we go to the design board reviews.

**Francis Aish//** Quite often yes. And we see the Partners say "I've seen ARD doing these cool things on this project". Also the project goes to the ARD and make sure you get some of that... That's quite nice because that really enforces the value. Having some feedback is quite well... People operate, they want to learn, they come to you and say "what's new"... They ask "can you show us"... We got the word out.

**Angelos Chronis//** When we start a work we start with the classics. We start within the strategy. If the problem hasn't been tackled before, we explain the team we can do it with or without this or this...

**Francis Aish//** It's a lot of consultancy role on a way.

**Angelos Chronis//** Sometimes it can even be like: the team come to us and ask for this and this, and sometimes we have to say we can't to it that time, you have to restrict your goals, or you can actually do that, it overcome your difficulties in an other way... (mime avec les mains des niveaux sur une échelle verticale)

**Francis Aish//** Sometimes it can be just a conversation with the team “look at this or this”, and some other times it can be six month long development to programming new things. It really varies, which is nice! (rire)

**Angelos Chronis//** We are... All goes done with discussions with the teams, like reviewing the process, we need this time to edit ... It’s a lot of our work.

**Aurelie de Boissieu //** H. Whitehead said he likes staying with the design teams when they explore new models and tools... Are you also doing that?

**Francis Aish//** Absolutely. I think... It very links to the teams: some teams are very involved, and some teams are quite confident when they come to us with a particular problem... it really varies.

**Angelos Chronis//** Sometimes we are with the team like continuously.

**Francis Aish//** The team are really integrated with us, and with the structure engineers, environmental engineers and the design teams. That works. That’s a very big push in the office. We now have two engineering groups in the office. About a year ago we add a big environmental company: Piers Heath Associates (30 people), a consultancy group we used to do a lot of work with. We also add Roger Ridsall-Smith Structure. Roger was one director at ARUP... So it brings a lot of structural engineers.

**Angelos Chronis//** These are two engineering we used to consult in the past. They do a lot of consultancy as we do, on a way... We work a lot with them.

**Francis Aish//** Part of our role is to help integrating the teams all together. We enable bringing different software in the process. That has been an interesting role the last year. That has been quite with intent in the office. I mean, we had worked very closely with this people before, but having them in-house is different. It requires more rigour but allows doing much faster circles. Having people in-house is pretty much ... you know... like if they lived in the office. You can call them the morning... you know! (rire) That’s good.

**Aurelie de Boissieu //** So you are now seven in the ARD, do you have full time integrated members in the Design Team?

**Francis Aish//** No. I mean in the older SMG we tried that. And that worked quite well some of the times. But with some groups it works less well. I think now because the general extend of Design Computation in the company is much better, we all less need to embed in teams. Often we work depth in the project for a long time. But because a lot of the routine stuff is... you know... There is a lot of walking and talking... We need less to be physically there. We can do it if necessary.

And because we are quite a small team we have to be quite careful on where are our resources. It’s nice to have a small team; we are all in a very same location. (rire)

**Aurelie de Boissieu //** Are you all architects in the ARD?

**Francis Aish//** No. We have three architect-engineers, includes Angelos. (rire) I am an aerospace engineer. Adam is in art and architecture. Tommy is a computer scientist and environmental engineer. Fong is an artist and architect... (rire) So we have a broad range of skills. It can be hard to find the right people to work here. We require very strong skills. Everyone is quite analytical, very bright.

**Aurelie de Boissieu** // Do you feel as an “Architects +”?

**Francis Aish**// We all got architecture skills. I worked here for several years. We all have plenty of experience, artistic and engineering.

**Angelos Chronis**// What we try to do is lend from other industries and putting it in architecture. (rire)

**Francis Aish**// Yes. We try do be multidisciplinary. It’s a broad... but...uhh.. It seems to work.

**Aurelie de Boissieu** // And in the team is there some training for the member?

**Francis Aish**// Yes. There is a lot of cross-training. All the in-house have general skills but we also have specific skills, so we try teaching people like the VBA or processing, etc.

**Angelos Chronis**// In most case we work in small group (2 or 3 people) in a project. So in that case there are at least 2 people who know the problem from both sides... And then they learn during holidays...

**Aurelie de Boissieu** // I don’t know if it’s too confidential, but is it possible to speak about one specific project?

**Francis Aish**// I think they are all quite different so we can only speak generally.

As a company we want to turn the design better. That’s a general statement but we want to make the building sort of... you know... better on the user point of view, better views out, better with more comfortable spaces. We also want to make some better environmentally... you know... using less energy, less material, make the more structurally efficient, making them cheaper, or making better quality for the same money... So typically we are broad to try pushing the boundaries, optimizing things and find new ways for doing things. So that can be very specific like analyse several problems, suggest a new form that has a very environmental performance or whatever. Very often nowadays what we try to do is combining multiple factors, what is the optimum form that treats with daylight versus solar gain, versus views, versus area for example. It is quite hard to do that process in your head or manually. So we can provide both the analyses that can confirm or deny certain thoughts ... and also you can optimize the form and use the computer to find the optimum. So it’s a kind of what we call performance driven design. We try to help the Design Team make very informed decision.

**Angelos Chronis**// Aurelie was also interested in the communication between the team and us, how, when,... How they contact us, how they explain what they need...

**Francis Aish**// Typically the Partner in charge of the project would come to us and say “I have this problem, I have this project..” So we get briefed on the project and on the background they have. We try to understand, to find some procedure we can reuse, we could suggest to them... Sometimes that happens in the middle of the Design Board Review “how can we solve the project?”...

It’s like a consultancy. It depends on the 3d model we have to analyse, or we might suggest new forms.

**Angelos Chronis**// Typically it will be like that: the team, the Partner, come to Francis and say “we have this problem in the project, we need to find a way to solve this, we need analysis, etc.”. Then Francis will define the strategy with this and talk to us, consult us. We discuss with the team to propose things and then at some point we say: “ok, this is how we gonna do, this is the plan of action”.

And then we explain the things we need, like 3d models with things like that or that, or like range of parameters. So we say to the team: we are going to do this, but we need from you that and that. Because even if the Design Teams realize more about the problems, that's happen also with the tool sometimes. Sometimes you realize just by talking about on how big your windows can be, on how is the range of parameters according to the design, you can figure out half of the problem. Just talking about the problem makes open eyes of the team and us.

Actually when we do the tool then we go back to the team and say "this is what you can do, you have this and this" and the team says "uhhh ok, actually we can't do that", so we restrict that. Sometimes the problem is not only finding a solution, sometimes it's more like building a kind of knowledge for the problem.

**Francis Aish//** Yes. It is definitely a dialogue. And also the designs evolve, and sometimes evolve very fast in this place! (rire) We have to be quite responsive. We have to find a process the team can use as well. Some of the projects we're involved in are quite long. It's a lot of concepts and cooptation the all way through... Things do evolve. We work at all stages of design. It's a lot of computation, a lot of concept stuff...

**Aurelie de Boissieu //** As the projects evolve always a lot, do you try do develop small models or small codes to reuse them?

**Francis Aish//** If we can yes. If we can reuse tool, we will say "this is a general problem I can solve it for little bit more effort". It depends on the project. Sometimes you can reuse tool or you can extended an existing tool. We do a lot of that, we take a tool and add some new functionalities. But some things are really specific or... uhh... In the end it is about delivering a solution, a building. In the end the goal is doing the building, in sketching, with computer or whatever... It's a hard thing to say but at the end you have to make the building. (rire)

**Angelos Chronis//** What you still interested in is Parametric Design?

**Aurelie de Boissieu //** Yes it is...

**Francis Aish//** We still do some parametric design. But we can't use just GC or just GH, it's broader. The skills are now far more common in the Design Teams. And a lot of the routine stuff can be handled by them. I am very happy for them to pick up...

**Angelos Chronis//** We mostly do some analysis on top of parametric modelling. But in most case the base for analysis is also a parametric model. And we develop sometimes things on GC that we can reuse to optimize...

**Francis Aish//** Yes. We are doing Parametric Design a lot as a Performance driven design. "Why should it be this shape?" is a question we have a lot from the Partners. That's not only a parametric model: it has to be a reason. It's more rational, it's less about pure geometry and more about the performance. It takes a while to get that stage, but I think now we have the tools, we have the experience.

The role does evolve. I have seen a lot of changes in the past thirteen years. The skills have changed. It is an interesting challenge. I think it is a good change. Skills increase. It happens what happened for the rendering. Twenty years ago rendering was a really specialist skill. Now rendering is

integrated in any office. Client expects that, and now client wants rendering, he wants movies, etc. I think now the same thing is happening for the parametric modelling. People expect that...

Do you have some more questions?

**Aurelie de Boissieu** // Well I have a lot of questions but it would be easier for me if we could talk about a specific project... For example can we speak about the ARD interpretative task? Or more about Parametric Design, do you share scripts on GC?

**Francis Aish**// Lets talk about GC. We do write components and features that can be reuse. Lot of them tends to be quite specific to a project. But they tend to be re-implemented in several projects. And also the employ of features in GC is not the easiest, whereas with VBA I can put a script on a single server, and everyone can see it. In GC it's hard. A lot of this is about the tool.

**Angelos Chronis**// Everyone have to copy and delete locally in every machine, that is the problem with GC. There is an example about that. I developed four different types of balcony. It was a case of features of external balcony that have to be applied to the building. But the use by the team of the features was difficult... exporting them, applying them... Sometimes it's not working.

**Francis Aish**// I think in GC, the concept of reimplementatation let a little bit to be desired ... I mean GH is better implemented. It is much easy to use even if the conception is less complete.

**Angelos Chronis**// Essentially what you have is both cases: if you build a parametric component it always depends on the geometry set up in you initial input. And if someone changes something, like the direction of a curve, then everything goes wrong. This kind of problem makes things difficult to work with. It is often better to build your one parametric model and you apply your component on, rather than the team does it. You need to have control on your initial input; you need to have a clean model.

**Francis Aish**// I think these tools require a lot of thoughts to set them up. It depends of the mindset. Some people have very logical process and some people have less of that. Some people want a simple result, and that a part of our role to take the complex problem, to simplify them to make them understandable. It does very a lot per project.

I think for your research it is better to know the general principles more than the specifics...

**Angelos Chronis**// You had also question on customization...

In most case you have to mass custom. Every project is different. Not only in terms of geometry and input but the goals also are different. You can't always apply the same methods. When you have set up the basic principles, you have to set up the interfaces: like how do I connect GC with this environmental analysis program, how do I connect, I don't know GC with something else... those links to the custom, to the project... Essentially our role today is also linking and interfacing analysis tools that weren't available to the Design Teams few years ago. We have to do this on a robust and easy way for the Design Team.

**Francis Aish**// Yes. I think also we do a lot of work in building work close: sequencing tools up, putting all together. The ends justify the means... You have to adapt the tool to the process, to the team, to the project...



**Aurelie de Boissieu** // That's a kind of collaboration tools...

**Francis Aish**// Yes... but also with the designers and the engineers ... we develop tools together, both for the project and more general purpose tools. That works quite well.

# ANNEXE 2 :

## Analyses

### Architecturologiques

## 2 Analyse de pratiques d'étudiants

### 2.1 « Concentricité et homothétie interne » de Thomas Perez et Nicolas Ruiz Gonzales (Studio P9)

Description du projet et du modèle paramétrique développé

Le projet « Concentricité et homothétie interne » propose un modèle générant des triangulations de surfaces convexes. Le système qui consiste en : 1- identifier le cercle dans lequel peut être inscrit la surface à trianguler (toutes les surfaces ne sont pas utilisable dans le système), 2- créer  $n_1$  cercles concentriques sur le plan xoy (cf. Figure 130 a), 3- créer  $n_2$  points réguliers sur les cercles, 4- créer un réseau de barres à partir des points précédents (cf. Figure 130 b), 4- projeter la géométrie de la triangulation sur la surface (cf. Figure 130 c).

Les étudiants ont appliqué ce système pour le cas d'une demi-sphère, choisie pour sa géométrie « pure » (cf. Figure 130 d).

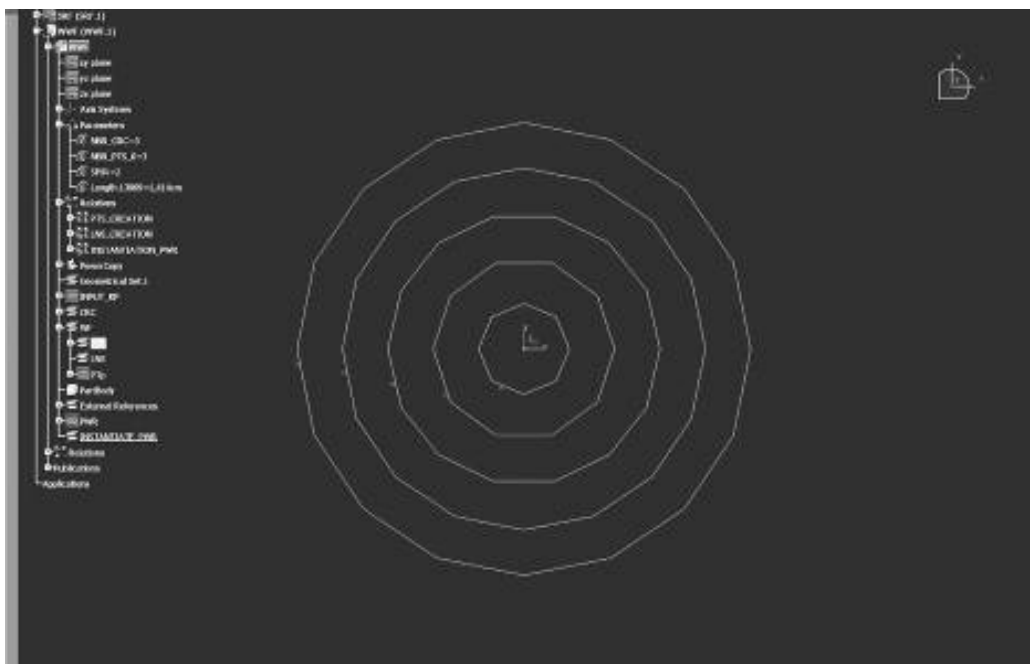


Figure 130a : Division géométrique de la surface



Figure 130b : Dessin de la triangulation

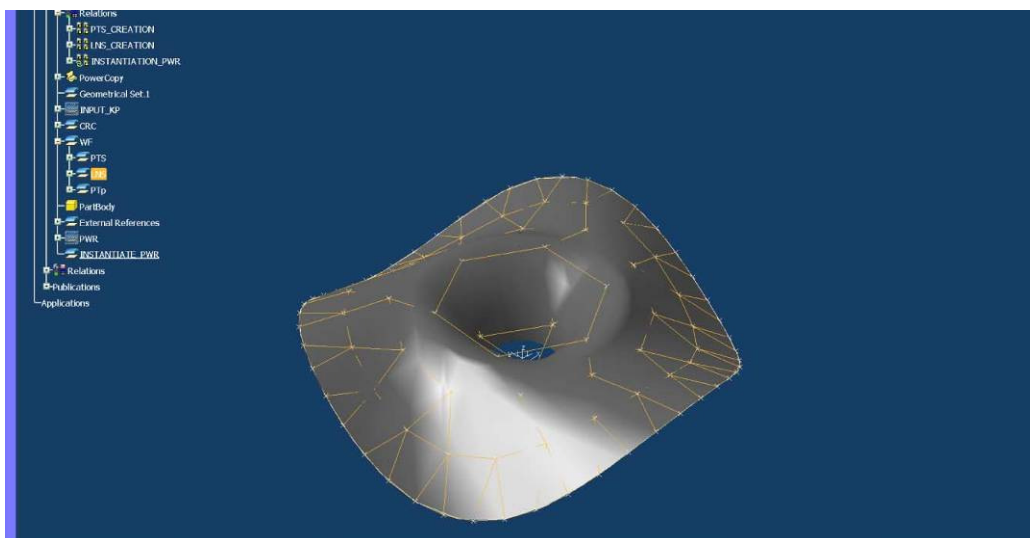


Figure 130 c : Projection de la triangulation sur la surface

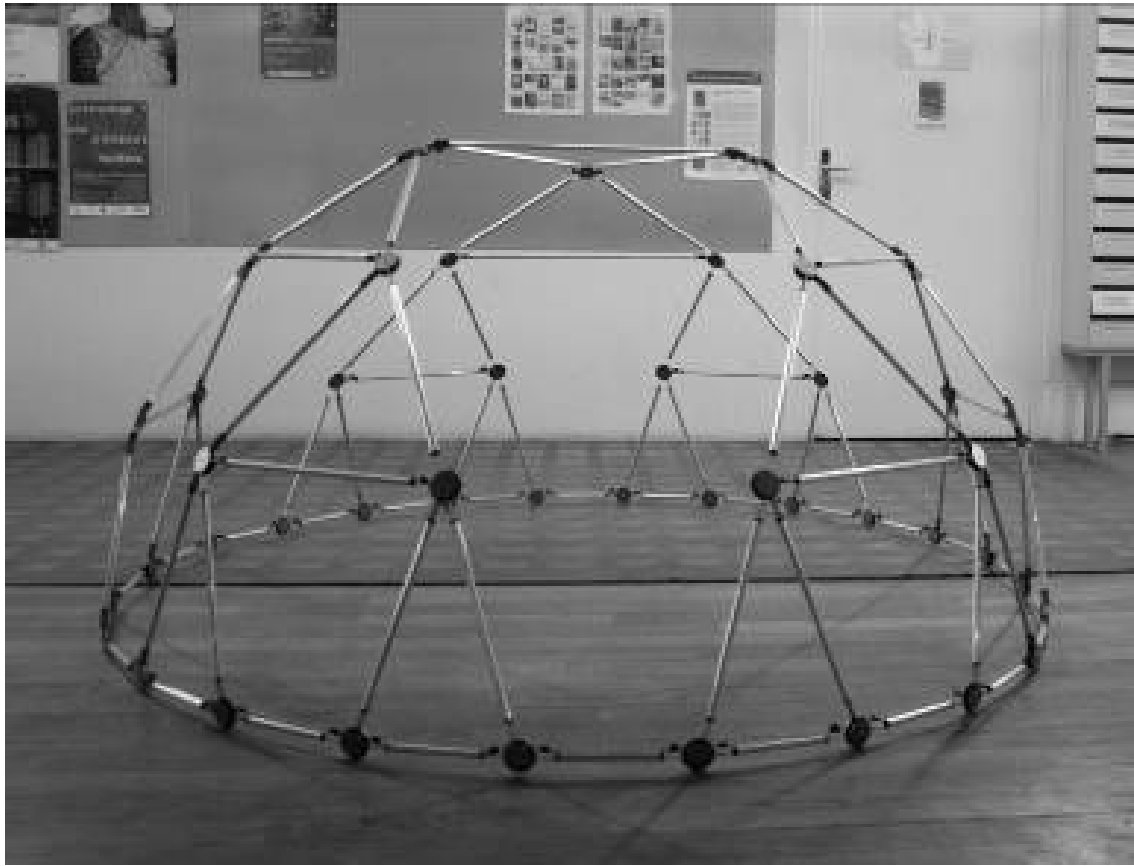


Figure 130 d : Instance construite par les étudiants

**Figure 130 : Images du projet « Concentricité et homothétie interne », sources : Thomas Perez et Nicolas Ruiz-Gonzales**

Description du processus de conception, des opérations cognitives impliquées et de leurs relations

| Opérations de Conception Architecturale  | Relation   | Opérations de modélisation paramétrique   |
|--|--|---|
| Echelle de niveau de conception : distingue la géométrie de la triangulation indépendamment du reste du projet |  |   |
| Conception de la triangulation suivant des échelles géométrique et technique                                   | <p style="text-align: center;">↔</p> <p style="text-align: center;">Traduction en géométrie paramétrique</p> | <i>Découpage</i> de la triangulation, selon des pertinences géométriques et techniques, en 1- conception d'une trame de points, 2- conception d'un réseau de barre reliant les points |
| Triangulation (embrayage) d'une demi-sphère choisie pour sa géométrie « pure » (échelle géométrique)           | <p style="text-align: center;">↔</p> <p style="text-align: center;">Traduction en géométrie</p>              | Instanciation du modèle en vue de le construire (embrayage) dimensionnée selon une échelle technique  |

|  |              |  |
|--|--------------|--|
|  | paramétrique |  |
|--|--------------|--|

Dans les principales opérations de mesures observées, on note que les échelles techniques et géométriques sont dominantes, non seulement dans la conception du modèle mais aussi dans son instanciation. Ce phénomène est peut-être lié aux attendus pédagogiques du studio P9 qui valorisaient la fabrication.

Les échelles géométrique et technique sont ici à la fois dominantes et structurantes.

Peut-on observer des « niveaux de conception » ?

On distingue ici deux niveaux de conception : un pour la conception du système paramétrique de triangulation (cf. Figure 130 a-b), l'autre pour son instanciation (cf. Figure 130 c-d).

Peut-on observer des opérations logiques ou pragmatiques liées à la modélisation paramétrique ?

Données non disponibles.

Principaux résultats :

|  |   |
|--|---|
| Conception d'un système de triangulation   | → niveau de conception de modélisation paramétrique<br>→ <u>échelles dominantes : technique et géométrique</u>                                  |
| Conception de l'instance « installation de triangulation d'une demi sphère »               | → niveau de conception d'instanciation  |
| Relation(s) entre conception du modèle paramétrique et conception de l'objet architectural | → <u>comme si les dimensions étaient les mêmes</u> pour le niveau de conception de la modélisation paramétrique et la conception architecturale |

## 2.2 « Blooming Surface » Studio P9

Description du projet et du modèle paramétrique développé

« Blooming surface » est un projet de couverture constituée par un module d'inspiration biomorphique (échelle de modèle). Cette couverture repose sur une structure développée par d'autres étudiants dans le cadre de l'atelier (échelle de voisinage). La possibilité d'hybrider des projets était proposée aux étudiants afin de croiser les contraintes et questionner les possibilités d'articulation entre les modèles.

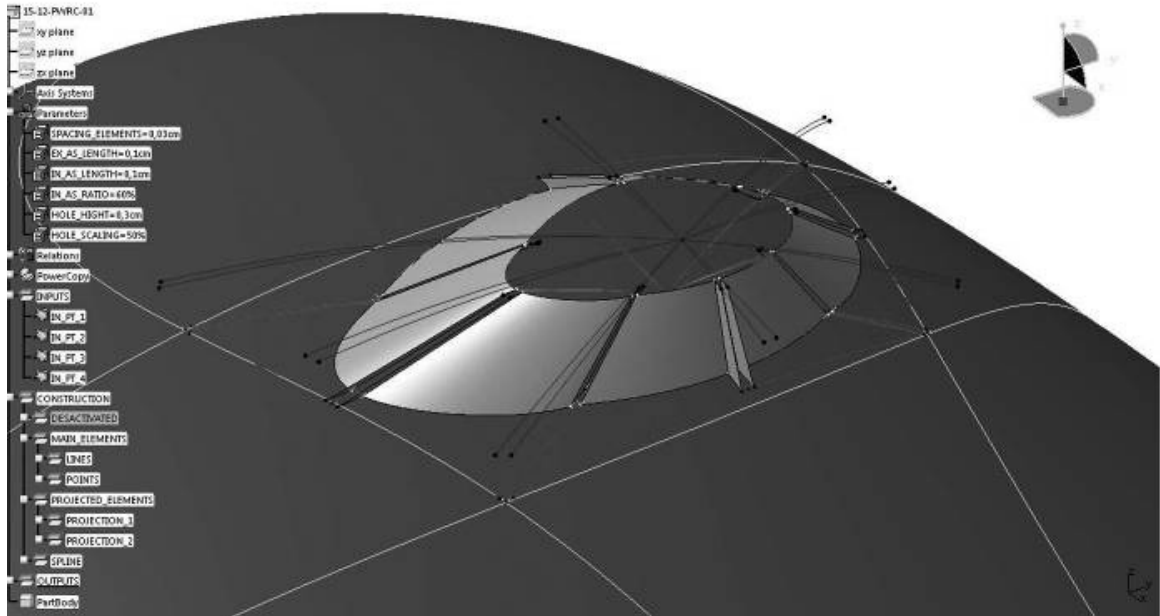


Figure 131 a : Powercopy du module, instanciée à partir d'une discrétisation de la surface de référence

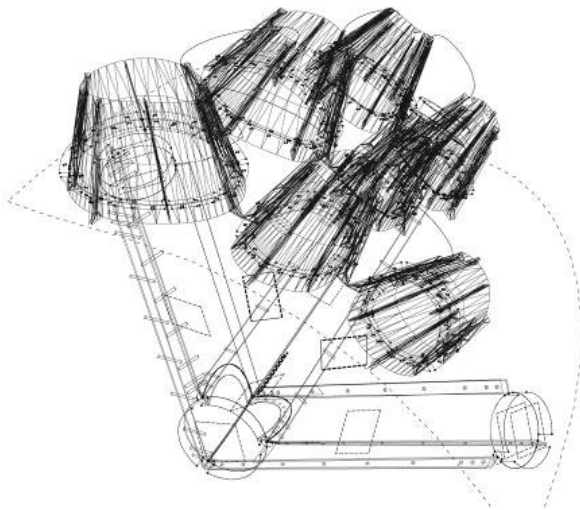


Figure 131 b : Hybridation des projets « Blooming surface » et « Arbre 615 »

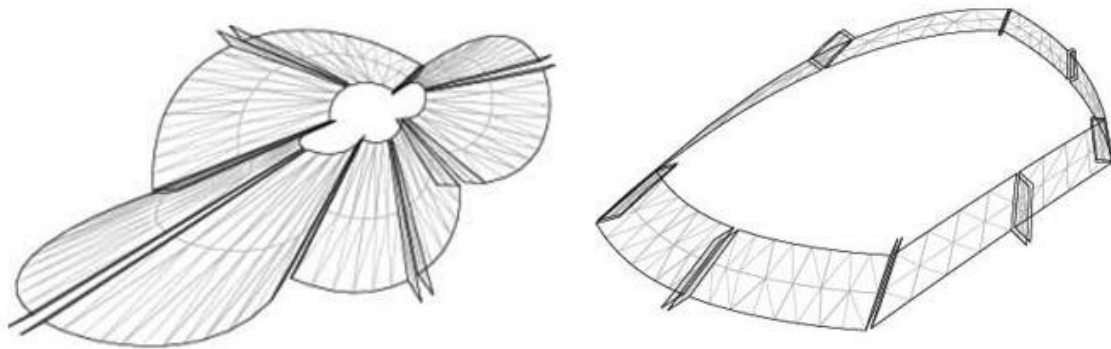


Figure 131 c : Exploration d'instances



Figure 131 d : Instance développée par les étudiants

**Figure 131 : Images du projet « Blooming Surface », sources : Maria Thala et Denice Martinez**

Après avoir décidé de partir d'une surface sur laquelle serait implanté un motif (découpage du modèle paramétrique), les étudiants ont exploré la possibilité de créer un motif en « anneau » (cf. Figure 126) (échelle géométrique). Puis elles ont interrogés la manière dont pourrait être construite cet anneau (échelle technique). L'embranchage vers une fabrication en feuille métallique a été très important dans le dessin du module. Ainsi dans les esquisses (cf. Figure 133) on voit les recherches de formes liées aux enjeux de la fabrication (échelle technique). Le module a finalement été conçu comme un assemblage de portions de surfaces développables (échelle technique).

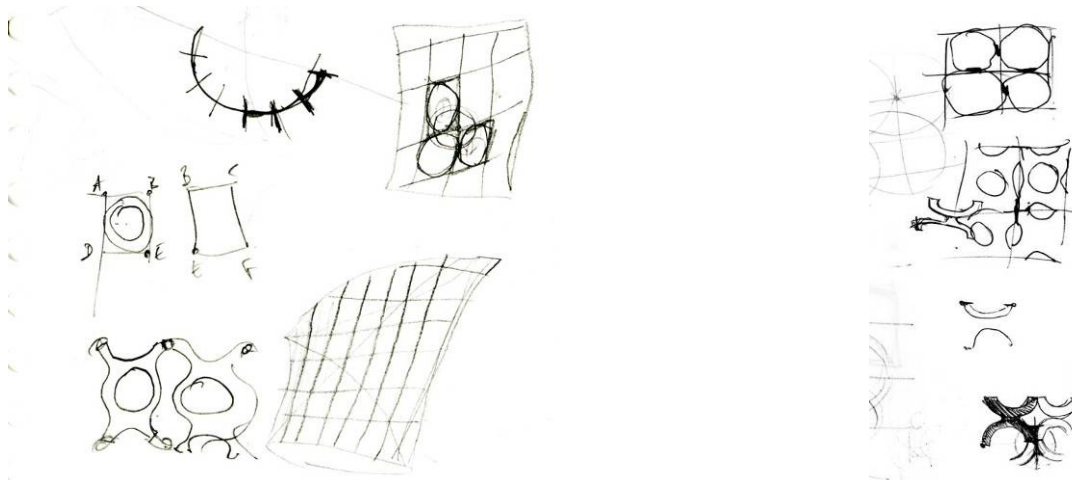


Figure 132 a : Premières esquisses du module à partir d'un « anneau »

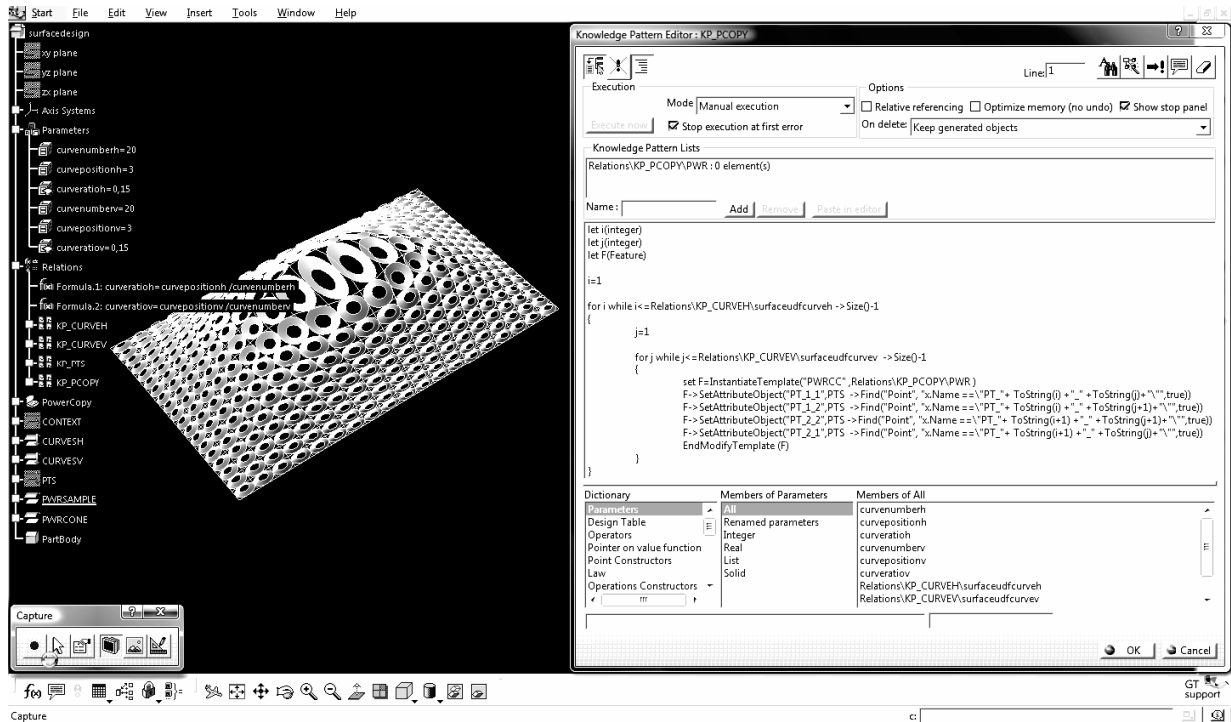


Figure 132 b : Premier modèle paramétrique

Figure 132: Premières explorations du motif en "anneau", sources: M. Thala et D. Martinez

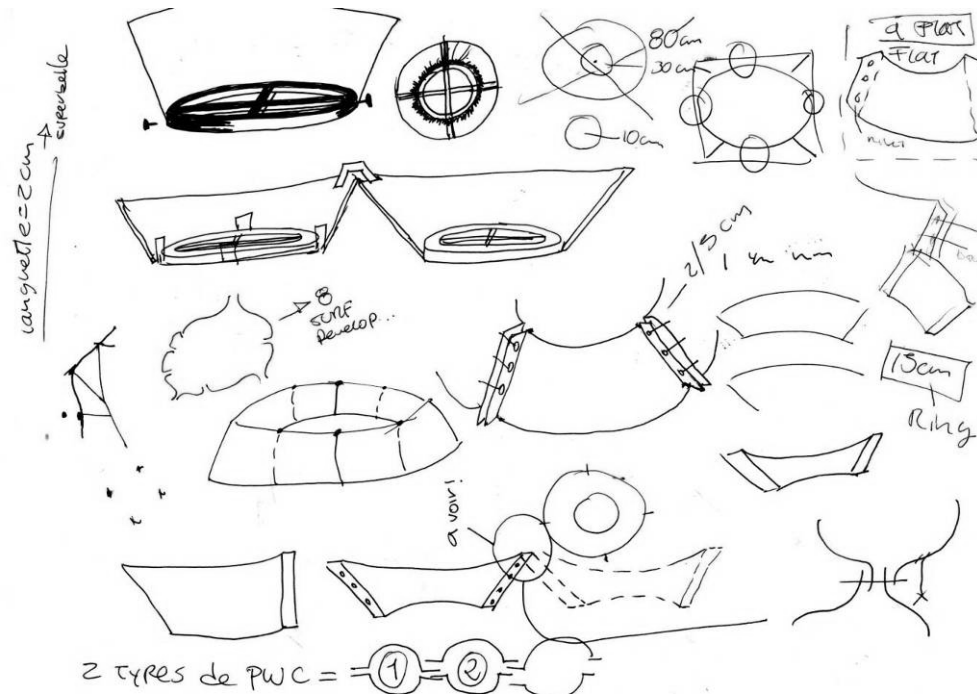


Figure 133: Esquisses montrant le processus de conception du module à partir d'une réflexion liée à sa construction, sources: M. Thala et D. Martinez



| Opérations de Conception Architecturale                                       | Relation   | Opérations de modélisation paramétrique  |
|---|--|--|
| Conception d'un motif en référence à un « fleurissement » (échelle de modèle) | <p style="text-align: center;">↔</p> <p style="text-align: center;">Traduction en<br/>géométrie<br/>paramétrique</p> | Conception d'un module de couverture suivant des échelles de modèle et technique |
| Constructibilité du projet (échelle technique)                                |  |  |

Les échelles de modèle et technique semblent dominantes.

Peut-on observer des opérations de modélisation paramétrique spécifiques ?

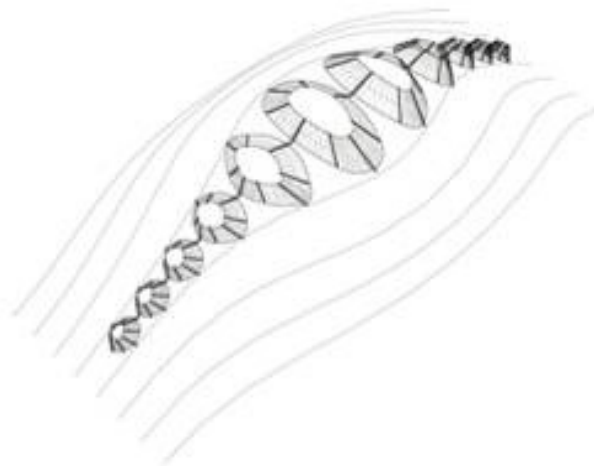
|                               |   |
|-------------------------------|---|
| Opérations de découpage       | Découpage du projet et du modèle en une surface sur laquelle est instancié un module  |
| Opérations de dimensionnement | Mise en relation de la surface et du module <i>via</i> quatre points ( <i>cf.</i> Figure 131 a), cette opération pose problème puisque la courbure de la surface n'est ainsi qu'approximée. |
| Champs de valorisation        | Hiérarchisation des contours de la surface, rendus plus important que sa courbure.  |

Peut-on observer des « niveaux de conception » ?

On observe un niveau de conception du modèle paramétrique pour le module *via* une powercopy, ainsi qu'un niveau de conception de l'instance selon des pertinences liées à la fabrication.

Peut-on observer des opérations logiques liées à la modélisation paramétrique ?

L'exploration des instances semble participer aux deux niveaux de conception : elle aide à tester la viabilité du modèle paramétrique ainsi qu'à visualiser la variété d'instances pouvant être générée (l'espace de solutions). La Figure 134 montre que des opérations de vérification ont été effectuées pour tester l'instanciation du modèle sur des surfaces différentes. Néanmoins si différentes instances ont été explorées, elles l'ont été sur des surfaces aux courbures très proches.



**Figure 134: Explorations d'instances, sources: M. Thala et D. Martinez**

Le module développé peut être instancié sur différents types de surfaces, même en cas de courbures très marquées. Cependant, les modalités d’ancrage des powercopies sur la surface ne prennent en compte que les quatre sommets générés par la discrétisation de la surface de référence. Cela ne permet pas de rendre compte de la courbure locale et générale de la surface, or, dans certains cas de fort changement de courbure, l’instanciation du module nécessiterait de penser la validité<sup>108</sup> des assemblages, de l’orientation du module ou de ses dimensions. Les conditions de validité de l’instanciation du modèle paramétrique seraient ici à questionner lors de la modélisation paramétrique : l’opération de validation si elle a été effectuée, reste ici partielle.

Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?

Données non disponibles.

Principaux résultats :

|   |   |
|---|---|
| Conception d’un module<br>« fleur »   | → niveau de conception de modélisation paramétrique<br>→ dominance des échelles techniques, de voisinage et de modèle<br>→ <u>importance de l’embrayage</u> (et de l’échelle technique) |
| Conception de l’instance<br>« installation d’une<br>blooming surface »                              | → niveau de conception d’instanciation  |
| Relation(s) entre<br>conception du modèle<br>paramétrique et conception<br>de l’objet architectural | → Traduction en géométrie paramétrique  |

### **2.3 « Les particules en actions », Hugo Houplain and Constance de Batz's (Studio P9)**

Description du projet et du modèle paramétrique développé

Le projet « Particules en action » propose un système paramétrique permettant de produire des modèles de structures à partir d’un système de « particules ». Si penser la fabricabilité de cette structure a été primordiale (une installation a été construite, cf. Figure 139) cette structure reste « générale » : elle est revendiquée par les étudiants comme pouvant être adapté à du mobilier comme à des structures d’immeubles de grande hauteur par exemple.

Le modèle se compose : 1- d’un « wireframe » contraignant le volume global de la structure à partir d’une partie plane (échelle fonctionnelle et géométrique) et de quatre hauteurs indépendantes (cf. Figure 135) (échelle géométrique), ce wireframe permet de définir une trame de points grâce à des variables indiquant le nombre de niveaux et de points (échelle géométrique); 2- d’un module

<sup>108</sup> Ce que permet Digital Project à travers les « lois » : il s’agit de contraindre l’instanciation d’une powercopy par un script, sur le modèle « if... then... ».

« particules », modélisé comme une powercopy, permettant de lier éléments verticaux et horizontaux à partir des particules définis en 1- (cf. Figure 136) et d'une réaction à un point attracteur (cf. Figure 136) (échelle technique et géométrique).

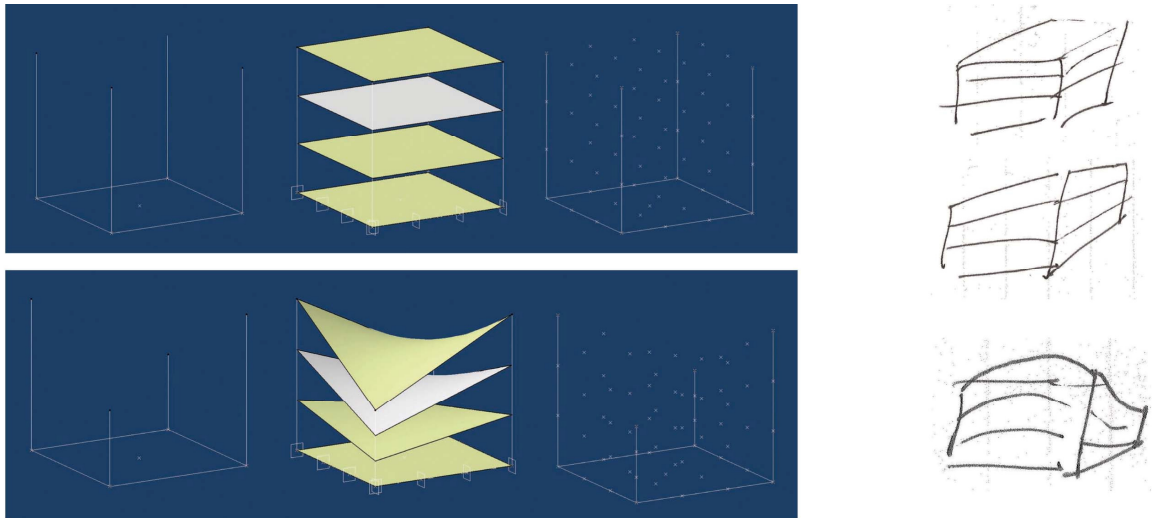


Figure 135: Schéma illustrant la construction du wireframe, source: H. Houplain et C. de Batz

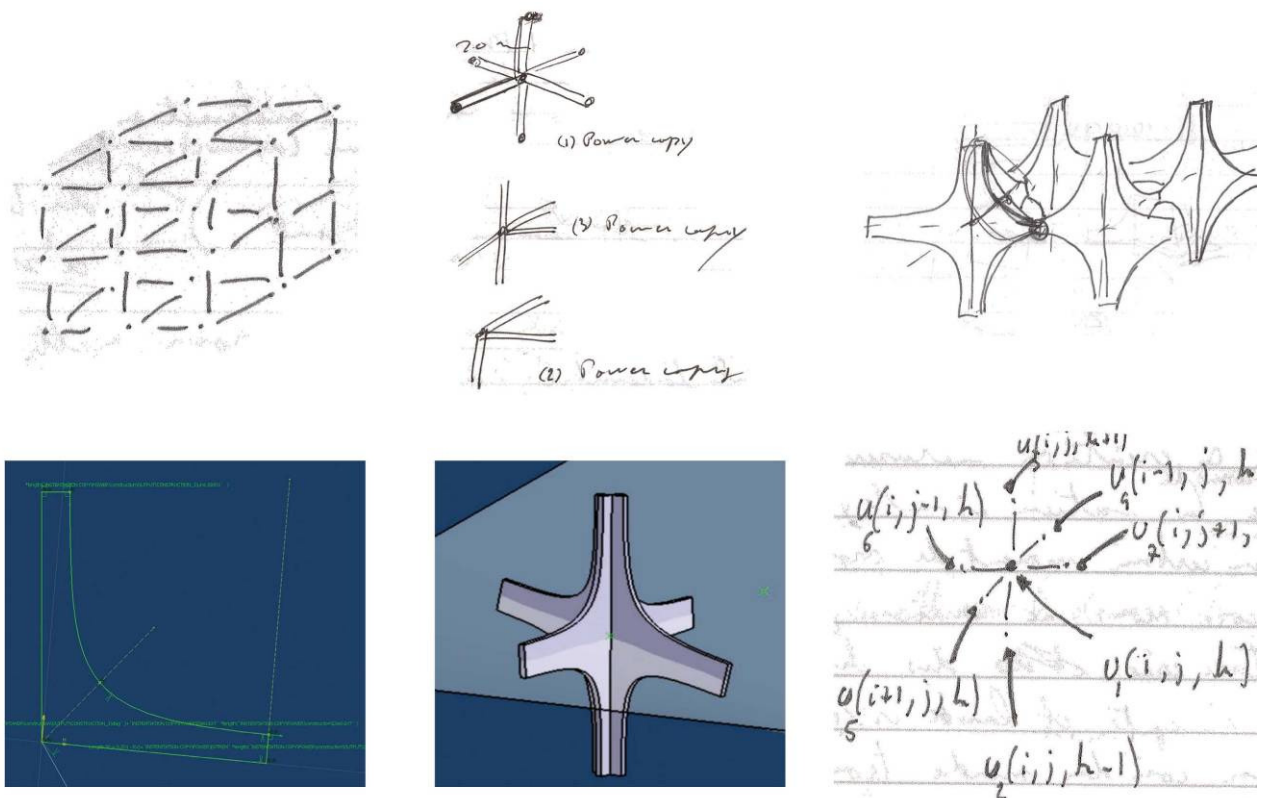
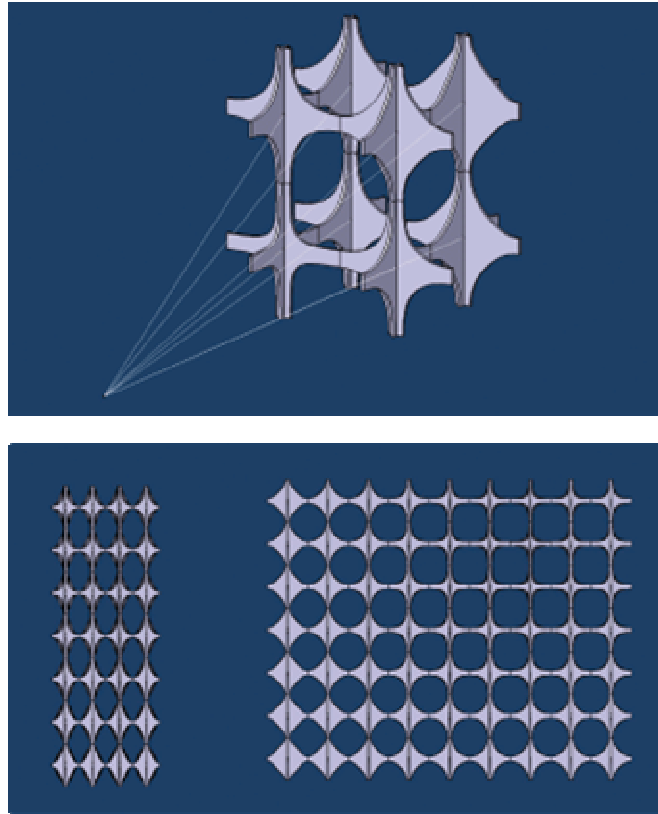
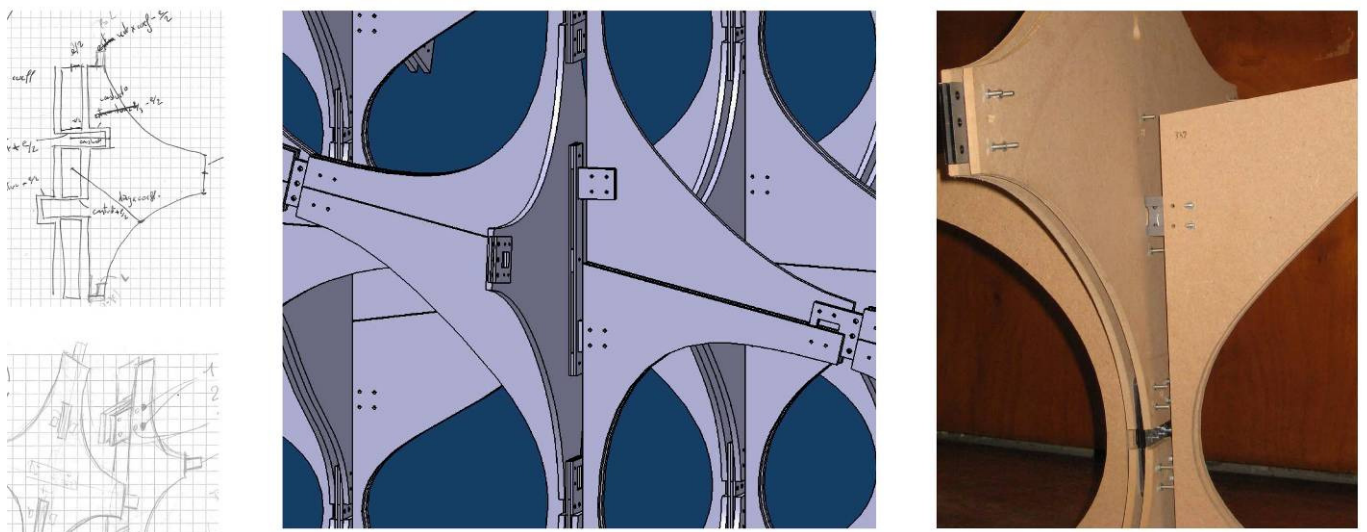


Figure 136: Images illustrant la définition du module, sources: H. Houplain et C. de Batz



**Figure 137: Images illustrant la réaction du module à un point attracteur, sources : H. Houplain et C. de Batz**

En lien avec la géométrie créée, des détails constructifs ont été conçus (cf. Figure 138 et Figure 139). Ces détails semblent avoir été conçus après la géométrie et à partir d'elle (épaisseurs, articulation, distinction entre plaques de medium et pièces mécaniques) (échelle technique et géométrique).



**Figure 138: illustration des enjeux de fabrication dans le projet "particules en actions", sources: H. Houplain et C. de Batz**



Figure 139: photo de l'installation "les particules en action", source: A. de Boissieu

Puis une instance « installation » a été conçue en vue d'être fabriquée le jour du rendu (échelle technique, fonctionnelle).

| Opérations de Conception Architecturale  | Relation  | Opérations de modélisation paramétrique  |
|--|---|--|
| Conception d'une trame régulière (échelle géométrique) définie par ses limites et dont cinq faces au moins sont planes (échelles fonctionnelle et technique) | <p style="text-align: center;">↔</p> <p>Sur-détermination<br/>(probable<br/>« relais »)</p>                   | Conception d'un « wireframe » permettant de générer la trame de l'installation (échelle géométrique) |
| Conception du système de « particule »<br>Echelle technique et géométrique.  | <p style="text-align: center;">↔</p> <p>Sur-détermination<br/>(probable<br/>« confusion des dimensions »)</p> | Conception d'un module grâce à une powercopy (échelle technique et géométrique)                      |
| Conception d'une instance de structure constructible pour l'installation (échelle technique) et rendant compte du travail effectué (échelle optique ?)       | <p style="text-align: center;">→</p> <p>infère</p>  | Conception d'une instance, choix des variables (échelle technique, fonctionnelle)                    |

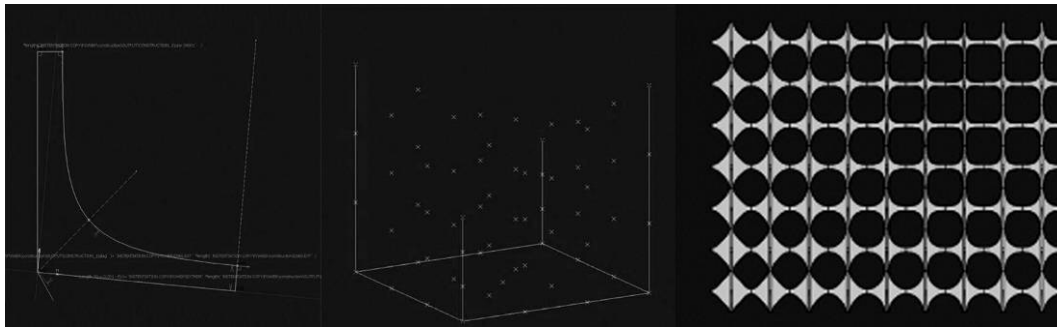
Ici les opérations de modélisation paramétriques et les opérations de conception architecturale semblent surdéterminer le wireframe et la powercopy « particule ». Les opérations de modélisation

paramétrique relative à l'instanciation semblent quand à elles en être inférées par les opérations de conception architecturale.

Peut-on observer des « niveaux de conception » ?

On observe un niveau de conception de modélisation paramétrique concernant la conception du module « particule » et la conception du wireframe. Un autre niveau est observable : c'est celui de l'instanciation du modèle paramétrique en un cas visant à être construit (cf. Figure 138 et Figure 139).

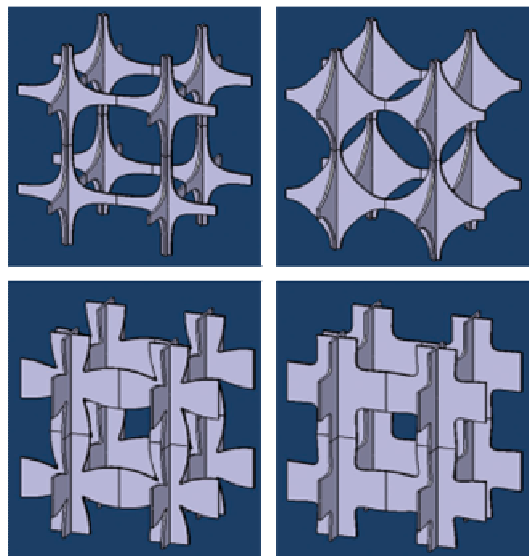
A partir du modèle paramétrique conçu, on peut imaginer une multitude d'instanciations possibles, variant en fonction d'échelles différentes : échelle de visibilité pour positionner le point attracteur, échelle humaine ou échelle fonctionnelle pour dimensionner le wireframe, etc.



**Figure 140: Schéma illustrant la succession des niveaux de conception, source: A. de Boissieu à partir des images fournies par H. Houplain et C. de Batz**

Peut-on observer des opérations de logique liées à la modélisation paramétrique ?

Au travers des représentations du projet fournies par les étudiants, on observe de nombreuses opérations de vérifications : 1- de la viabilité du module en fonction des modifications des variables (cf. Figure 141), 2- de la viabilité des évolutions conjointes du wireframe et du module (cf. Figure 142).



**Figure 141: Evaluation des modules, sources: H. Houplain et C. de Batz**



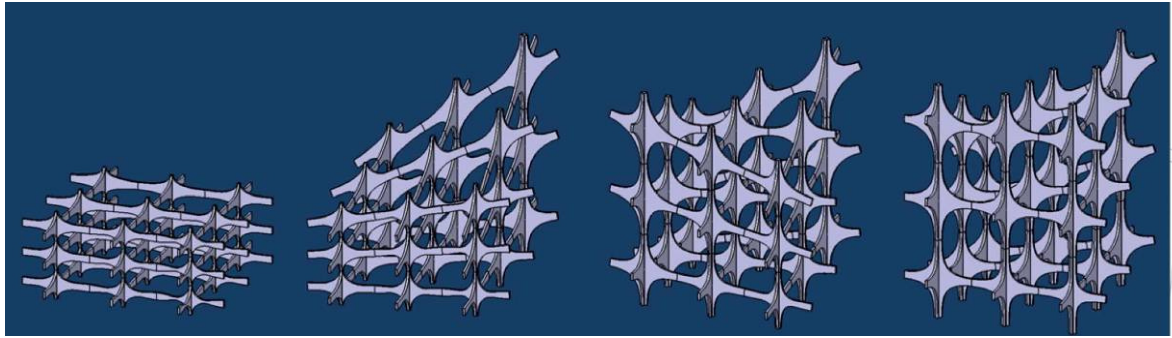


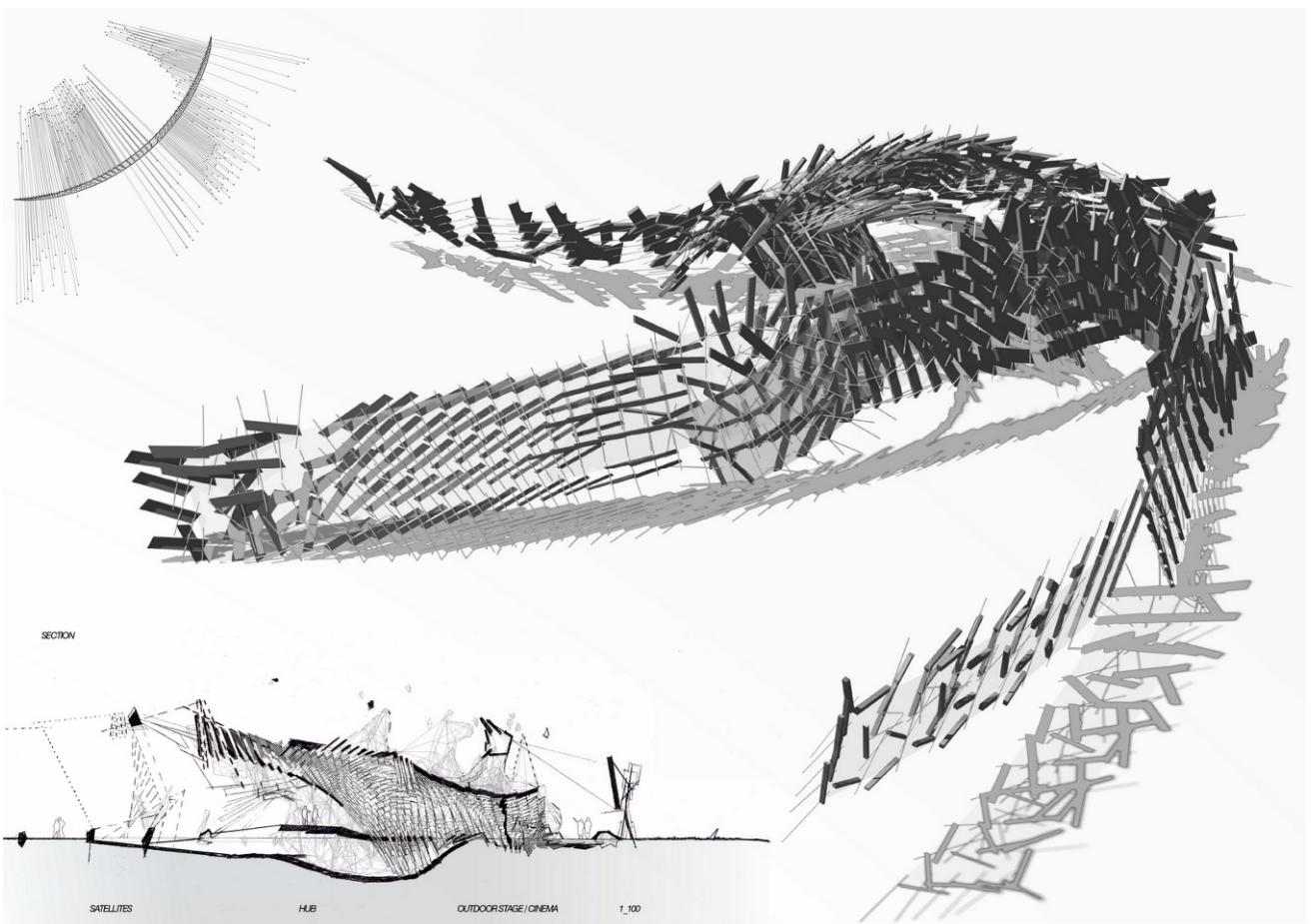
Figure 142: Evaluation du wireframe, sources: H. Houplain et C. de Batz

Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?

Données non disponibles.

## 2.4 « Radical Reproduction » de Daniela Kroenhert et Adam Orlinsky (Crossover Studio )

Le projet « Radical Reproduction » (cf. Figure 143) répond à un programme de « pavillon innovant » implanté sur le site du parc S. Freud à Vienne.



**Figure 143: perspective et coupe du projet "Radical Reproduction", sources: Daniela Kroenhert et Adam Orłinsky**

Description du processus de conception, des opérations cognitives impliquées et de leurs relations

La particularité de la conception décrite et revendiquée par les étudiants est de s'organiser en plusieurs phases : 1- la conception de systèmes de calepinages (cf. Figure 144a), 2- la conception de surfaces constituant les espaces du pavillon (cf. Figure 144b), et 3- l'instanciation des calepinages sur les surfaces (cf. Figure 144c).

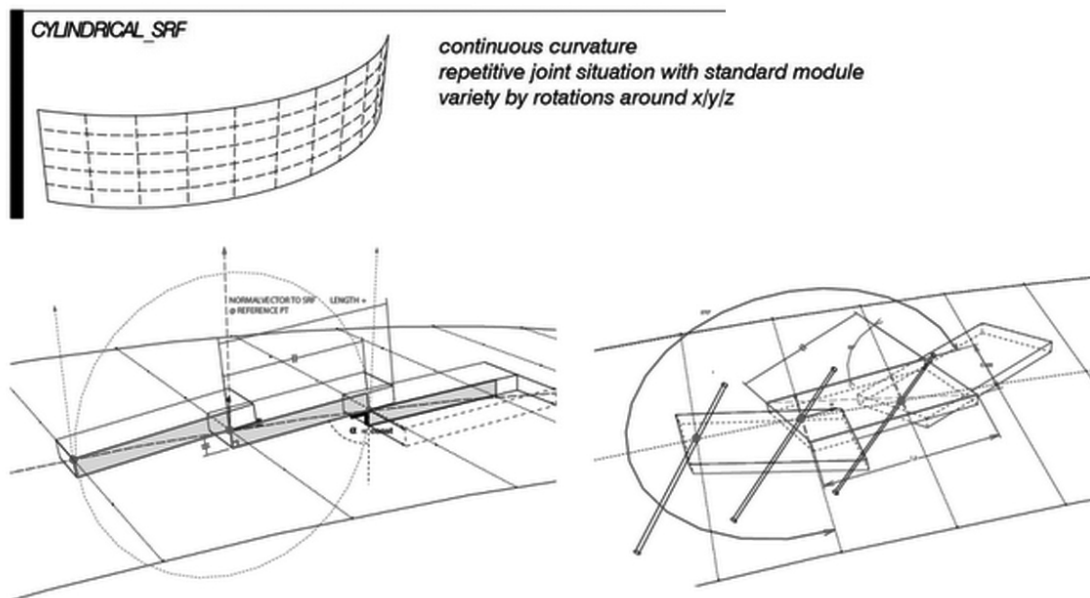


Figure 144 a : Typologie de calepinage en fonction de la nature des surfaces, ici pour une portion de cylindre



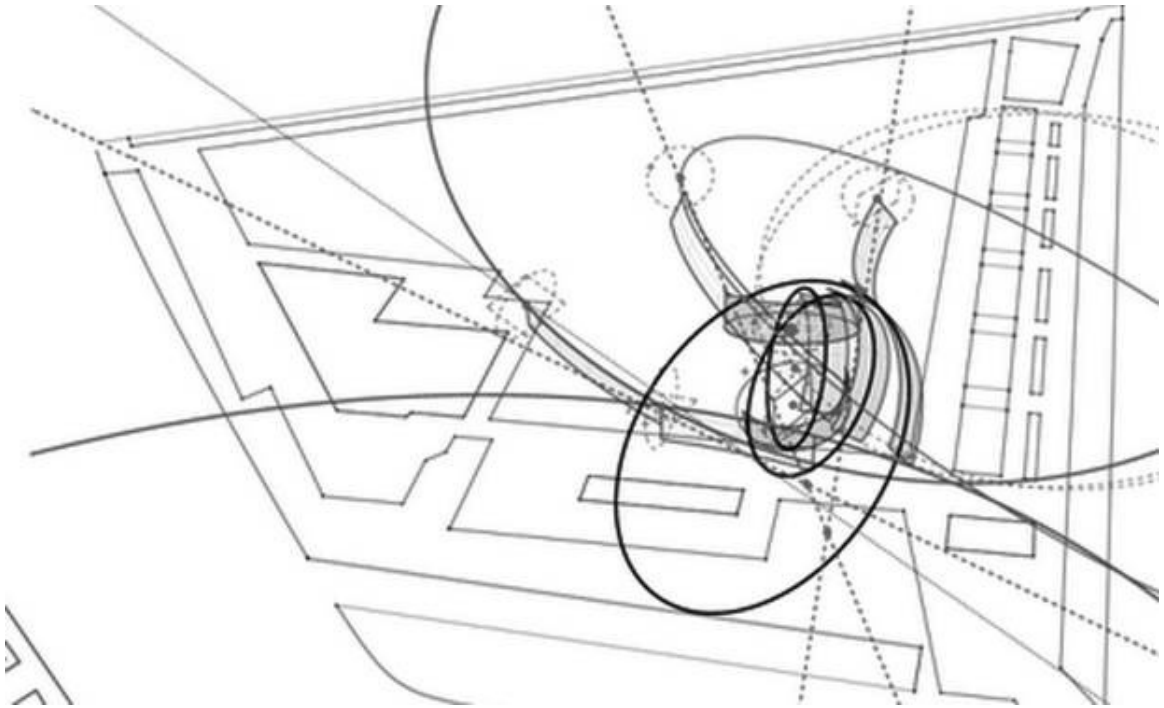


Figure 144 b : Dessin des surfaces du pavillon « according to urban parameters »

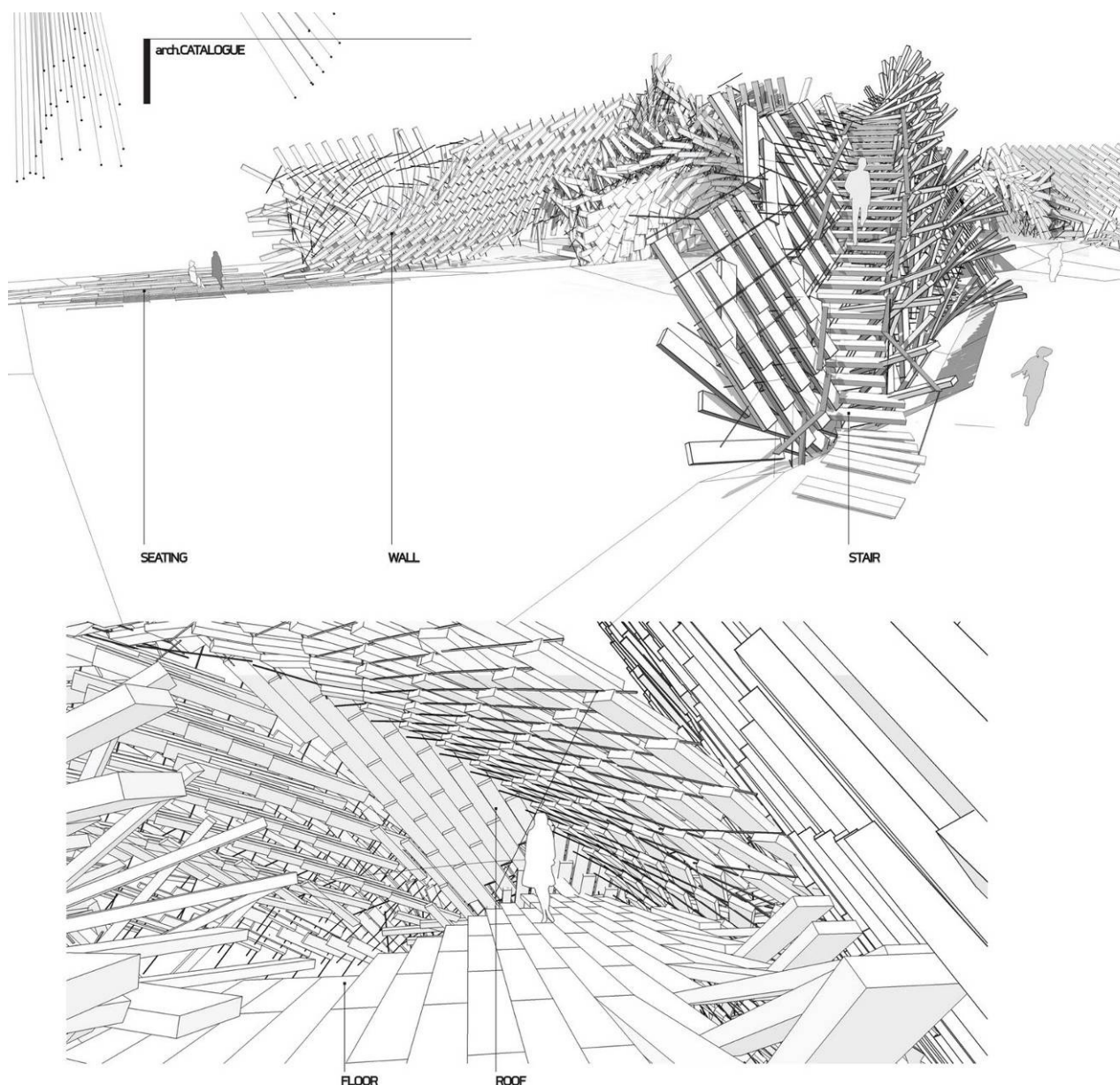


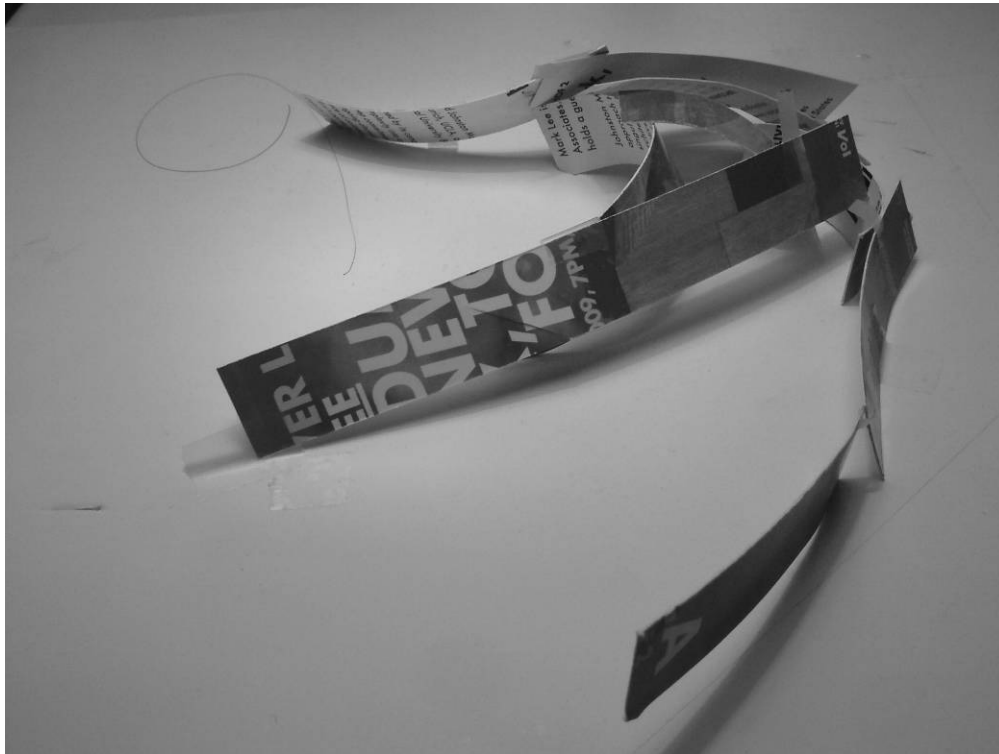
Figure 144 c : Instanciation des calepinages sur les surfaces du pavillon

**Figure 144 : « Radical ReReproduction » Daniela Krohnert et Adam Orlinski, Crossover Studio (Université des Arts Appliquées, Vienne)**

Le processus s'initie par la recherche de différents types de calepinage de tasseaux de bois (cf. Figure 144.a). Chaque type de calepinage est adapté à la nature des surfaces auxquelles les assemblages sont destinés (surfaces planes, cylindriques, coniques ou à double courbure inversée). Des évaluations de « surfaces type » grâce au moteur de calcul par éléments finis intégré dans Digital Project ont été effectuées pour permettre aux étudiants de définir les contraintes s'appliquant sur les assemblages (échelle géométrique, échelle technique).

Parallèlement au temps 1, les étudiants ont pensé un ensemble de surfaces répondant au site (cf. fig. Figure 144.b). Ces surfaces ont été conçues grâce à des expérimentations sur des maquettes physiques (cf. Figure 76). La conception de ces surfaces a été définie en fonction du voisinage avec les

volumes de la place (échelle de voisinage) et des flux des passages des piétons (échelle fonctionnelle). Le site, s'il intervient dans la conception des surfaces (cf. Entretiens, discours du jury et représentation de rendu), n'est pas pris en compte dans le modèle paramétrique : il est représenté (cf. fig. Figure 144.b) mais n'est pas lié paramétriquement aux surfaces (une transformation du tracé du site n'influe pas sur les surfaces). Il ne semble pas y avoir d'opérations de modélisation paramétrique liée à la conception des surfaces constituant le pavillon.



**Figure 145 : Photo d'une maquette d'étude, source: Adam Orlinsky**

Le processus de conception se conclut par l'instanciation du système de calepinage pensé *a priori* (lors du temps 1) sur les surfaces définissant le pavillon (conçues en 2). Ce travail d'instanciation se compose de deux temps : une instanciation automatisée des typologies de calepinages, puis une modification de certains assemblages (cf. Figure 146) en fonction des vues voulues (échelle de visibilité), des usages (échelle fonctionnelle), de l'effet de « swarming » souhaité (échelle optique) et de l'ensoleillement (échelle géométrique) (cf. Figure 144 c). L'instanciation n'est donc pas ici systématique, mais est le lieu d'opérations de conception variées.

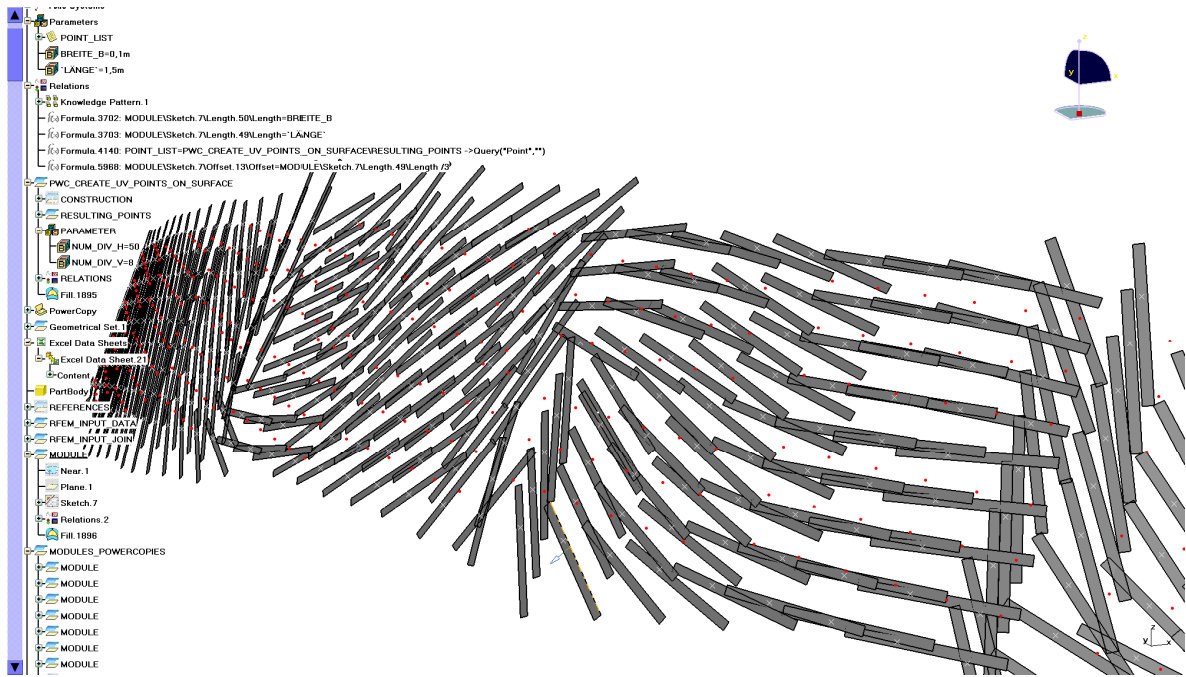
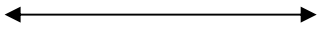

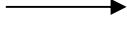


Figure 146 : capture d'écran du modèle paramétrique développé pour le projet "Radical Reproduction", source: Adam Orlinsky

| Opérations de Conception Architecturale  | Relation   | Opérations de modélisation paramétrique  |
|--|--|--|
| La conception des calepinages relève principalement de mesures géométriques et techniques : elles sont liées aux capacités techniques des assemblages de tasseaux de bois et à la géométrie des surfaces.                                | <br>Traduction en géométrie paramétrique<br><br>On peut émettre l'hypothèse que savoir qu'une power-copy devait être créée à amené les étudiants à penser spécifiquement en terme de système et d'entrée et de sorties claires. | Les caractéristiques de ces calepinages sont intégrées dans des « powercopies » via la définition de règles de comportements de celles-ci.   |
| Pour la conception de ces surfaces le voisinage avec les volumes de la place (échelle de voisinage) et les flux des passages des piétons (échelle fonctionnelle) ont été pris en compte.   | <br>Est suivi de<br><br>Disjonction entre les opérations de modélisation et les opérations de conception du temps 2.  | La modélisation des surfaces ne prend en compte que des pertinences techniques : elles sont représentées comme des portions de cylindres. Elles répondent donc à l'échelle technique mise en œuvre lors du temps 1, mais ne répondent pas aux échelles de voisinage et fonctionnelle mises en œuvre lors du temps 2. |
| Mise en place du calepinage en fonction des vues voulues (échelle optique), de rôles tels que « mur », « plafond », « banc (cf. Figure 144 c) (échelle fonctionnelle) en fonction des effets de « swarming » créés (échelle de modèle ?) | <br><br>Est suivi de<br>-Échelles de modèle, technique, fonctionnelle et optique<br><br>Détermination de l'instanciation de la powercopy par les opérations de  | L'instanciation a eu lieu en fonction de la nature des surfaces (échelle technique), tel que défini lors du temps 1, mais aussi en fonction des opérations architecturales liées à la définition des effets voulus par le mouvement des tasseaux de bois (échelle optique, de modèle,                                |

|  |             |                 |
|--|-------------|-----------------|
|  | conception. | fonctionnelle). |
|--|-------------|-----------------|

Peut-on observer des « niveaux de conception » ?

Dans un premier temps, des opérations ont permis de donner de la mesure au système génératif de calepinage. Dans un deuxième temps, des opérations cognitives de conception ont donné de la mesure à l'instance « pavillon » de ce système. L'analyse montre une différenciation explicite (et revendiquée) de la conception du système paramétrique par rapport à son instanciation. Il semble que nous soyons ici en présence de deux niveaux de conception : celui du modèle paramétrique et celui de l'instance. Il est intéressant de noter que les pertinences dont relèvent les deux niveaux de conception ne sont pas les mêmes. Dans le premier temps, la conception du système paramétrique relève principalement de mesures géométriques et techniques : elles sont liées aux capacités techniques des assemblages de tasseaux de bois et à la géométrie des surfaces. Dans le deuxième temps, l'instanciation de l'objet « pavillon » (temps 2 et 3) fait appel à des pertinences plus variées, liées à l'échelle humaine, aux usages, ainsi qu'à la vue sur le parc et à l'ensoleillement.

Peut-on observer des opérations de modélisation paramétrique spécifiques ?

|                               |  |
|-------------------------------|--|
| Opérations de découpage       | <p>Dans les multiples niveaux de découpage possible on peut ici mettre en avant le découpage explicite des niveaux de conception et de modélisation (échelle de niveaux de conception ?)</p> <p>La décision de discrétiser le modèle des calepinages par tasseaux de bois unique (et non par rangée par exemple) chacun possédant ses caractéristiques (orientation, point d'ancrage) peut être considéré comme une opération de découpage.</p> <p>Le modèle paramétrique n'étant pas disponible il n'est pas possible d'analyser très avant le découpage de celui-ci sur la seule base des captures d'écran</p> |
| Opérations de dimensionnement | <p>L'association des power-copies aux surfaces <i>via</i> des trames orthonormée de plans tangents est une mise en relation de la dimension calepinage avec les intentions architecturales (de vue, de fonctionnement du pavillon, <i>etc</i>).</p>  |
| Champs de valorisation        | <p>L'utilisation de la courbure des surfaces <i>via</i> les plans tangents pourrait être considérée comme une hiérarchisation de la courbure. Cette valorisation de la courbure des surfaces est liée à l'utilisation récurrente de l'échelle technique dans la conception du pavillon et de son modèle paramétrique.</p>  |

Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?

Données non disponibles.

Peut-on observer des opérations de collaboration liées à la modélisation paramétrique ?

Données non disponibles.

## 2.5 « Pavilion » de Philip Hornung (Xover Studio)

Description du projet et du modèle paramétrique développé

Philip Hornung dans son projet « Pavilion » a traité la volumétrie globale de son pavillon avec un modèle paramétrique. Le projet consiste en une recherche d'agencements de volumes et de surfaces en fonction d'un espace principal et clos, autour duquel s'organisent les autres espaces (cf. Figure 147 a). Suite à une difficulté rencontrée sur Digital Project lors de la représentation de la structure de l'édifice, l'étudiant a décidé de poursuivre son travail sur Rhinocéros (cf. Figure 147 c) (opération d'appropriation de l'outil). Néanmoins, l'étudiant a su maîtriser son modèle paramétrique pour hiérarchiser les mesures de son volume. Son modèle paramétrique est structuré autour d'un paramètre principal qui est la « boîte » contenant son espace majeur (cf. Figure 147 et Figure 150).

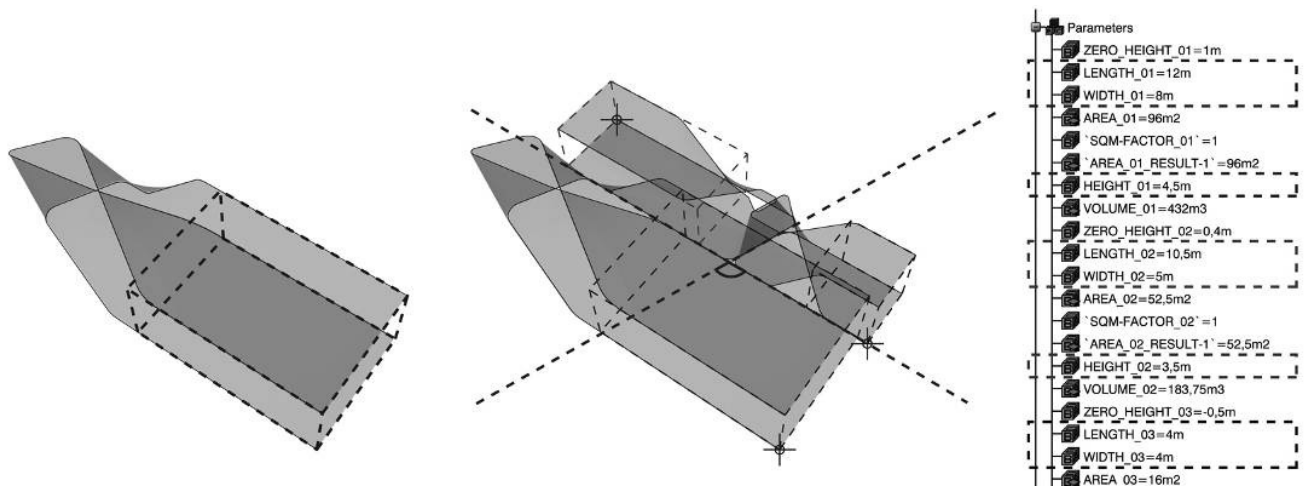


Figure 147 a : Hiérarchisation des paramètres autour d'un volume principal

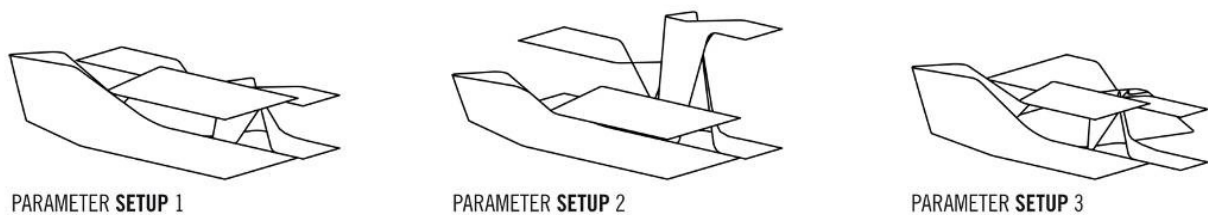


Figure 147 b: Exploration d'instances



Figure 147 c : Instance « pavillon » développée par l'étudiant

**Figure 147 : Images extraites du Rendu « Pavilion » de Philipp Hornung pour le Crossover Studio 2010, source : Extraits effectués par A. de Boissieu à partir des images de Philip Hornung**



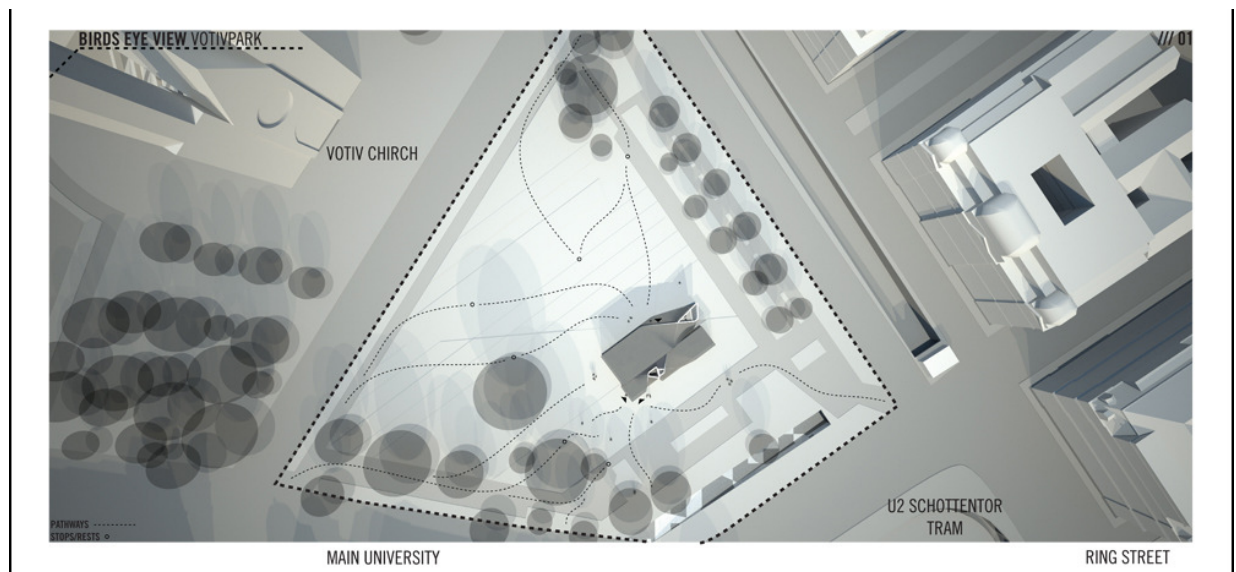


Figure 148 : Implantation du projet dans le site, source: Ph. Hornung

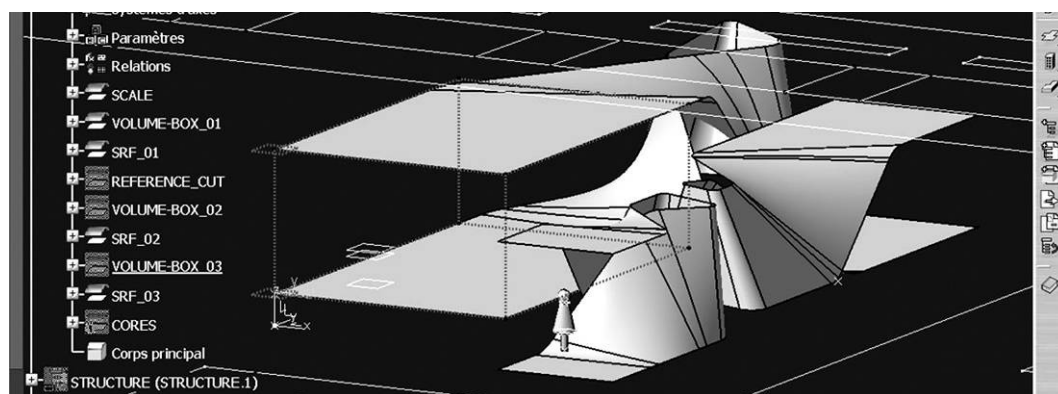


Figure 149 : Capture d'écran du modèle paramétrique final du projet « Pavillon » de Philipp Hornung, source : Capture effectuée par A. de Boissieu à partir du modèle de Philip Hornung

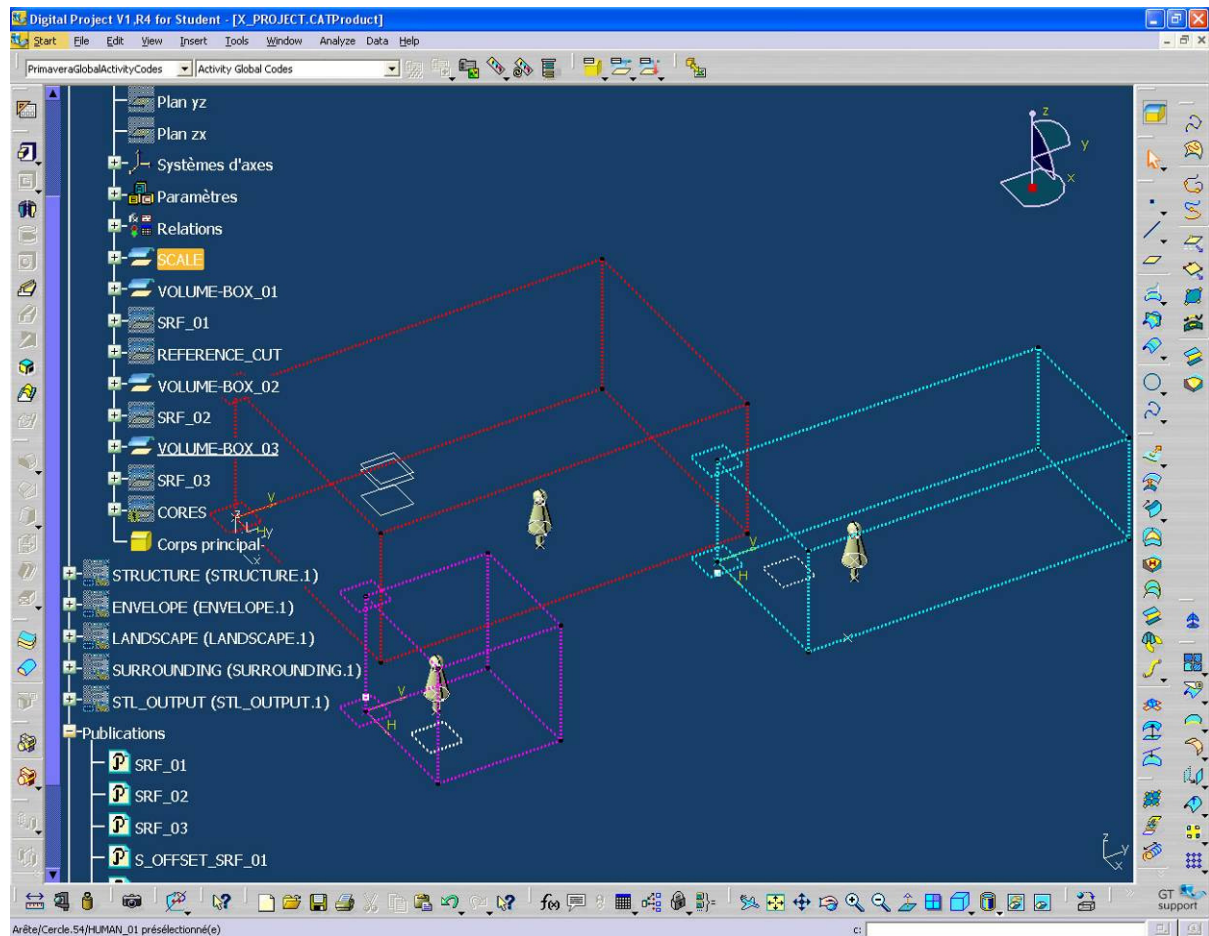
Le modèle paramétrique est constitué de :

1- une organisation de dimensions proportionnelles entre deux volumes (cf. Figure 147 et Figure 150), 2-une construction topologique qui relie le sol et le plafond de chaque volume en une surface « pliée » (cf. Figure 149), 3- une hiérarchie des volumes et de leurs dimensions en fonction de l'échelle humaine et des fonctions prévues pour les espaces (cf. Figure 149).

Ce modèle paramétrique définit un domaine de solutions que peut explorer le concepteur pour définir son instance « pavillon » (cf. Figure 147 b).

Dans la Figure 149 montrant le modèle paramétrique, on observe les trois surfaces « pliées » ainsi qu'un parallélépipède contenu dans le plus grand des trois volumes. Cette « boîte » est le volume définissant l'espace principal du projet, à partir duquel les deux autres volumes sont définis, ainsi que les surfaces pliées. Dans le modèle paramétrique (cf. Figure 150) on observe également la représentation stylisée d'un personnage et du tracé du contexte urbain. Si on peut émettre l'hypothèse

que ces derniers éléments ont participé à la visualisation du projet et donc à sa mise à l'échelle, ils ne sont néanmoins pas utilisés en tant que tel comme paramètre dans le modèle paramétrique.




**Figure 150: capture d'écran du modèle "Pavilion", source: image d'A. de Boissieu à partir du modèle de Ph. Hornung**

### Description des opérations cognitives impliquées et de leurs relations

| Opérations de Conception Architecturale  | Relation                                  | Opérations de modélisation paramétrique  |
|--|---|--|
| Mise en relation du rôle d'un espace avec ses dimensions<br>(Echelle fonctionnelle et symbolique formelle) | ↔<br>Traduction en géométrie paramétrique | Opération de mise en relation des dimensions de trois parallélépipèdes en fonction d'un ratio. Leurs hauteurs restent indépendantes.   |
| Echelle fonctionnelle : la hauteur doit rester maîtrisée ?   | →<br>Est suivi de                         | Opération de hiérarchisation des hauteurs des trois « boîtes » définissant les espaces (cf. Figure 150), ces hauteurs étant des variables indépendantes.   |
| Dimensionnement des espaces en fonction de l'échelle humaine (cf. Figure 149)                              | →<br>Est suivi de                         | Représentation de personnages pour assister la visualisation de l'échelle (cartographique) du modèle, mais pas d'utilisation explicite de cette « échelle humaine » dans la construction des relations paramétriques |



|  |   |   |
|--|---|---|
| Détermination des dimensions et du positionnement des espaces en fonction des flux piétons (cf. Figure 148)<br>(Echelle fonctionnelle) | <br>Est suivi de | Représentation du tracé urbain pour assister le positionnement des volumes dans le modèle, mais ce tracé n'est pas utilisé dans la construction des relations paramétriques générant le pavillon lui-même |
|  |   | Instanciation de dimensions des volumes pour assurer l'échelle fonctionnelle  |

Peut-on observer des « niveaux de conception » ?

On observe un niveau de conception relatif à la conception paramétrique (cf. Figure 147a) et un niveau de conception relatif à la recherche de l'instance « pavillon » (cf. Figure 147 b).

Peut-on observer des opérations de modélisation paramétrique spécifiques ?

|                               |   |
|-------------------------------|---|
| Opérations de découpage       | Opération de découpage du pavillon en trois volumes dirigé par des parallélépipèdes (le « ruban » est déduit)                             |
| Opérations de dimensionnement | Dimensionnement des trois volumes du pavillon les uns par rapport aux autres (échelle fonctionnelle et symbolique formelle)               |
|                               | Opération de dimensionnement du ruban en fonction du parallélépipède : le ruban est déduit des polygones générées par le parallélépipède. |

Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?

On peu observer une opération d'appropriation de l'outil : en effet quand l'étudiant a éprouvé des difficultés avec DP il a changé de logiciel. Il a ciblé l'usage de DP sur les points (générer la géométrie globale) pour lesquels il y a trouvé un intérêt (contrôle du dessin et performance de la représentation).

Peut-on observer des opérations logiques liées à la modélisation paramétrique ?

Données non disponibles.

## 2.6 « Fold » de Siim Tuksam (XOver Studio)

Description du projet et du modèle paramétrique développé

Le travail de Siim Tuksam a porté sur une recherche géométrique de module visant à reconstituer une surface convexe grâce au pliage d'une surface développable. Le projet s'inspire des origamis et crée un module (cf. Figure 151a) répété sur la surface du pavillon à partir de deux axes (cf. Figure 151b). Si Digital Project a permis à l'étudiant de représenter la totalité de son pliage (cf. Figure 151c),

il faut néanmoins noter que la powercopy élaborée pour son module ne s’instancie pas quand la courbure de la surface de référence est trop forte (cf. Figure 154). L’instanciation de la powercopy de pliage nécessite une intervention de l’utilisateur : le modèle entier n’est donc pas paramétrique. En effet, si l’on modifie la surface de référence ou les axes structurant l’instanciation, les powercopies sont à réadapter manuellement<sup>109</sup>. Le travail de modélisation prenant vraiment en compte les mécanismes paramétriques du modelleur semble avoir été celui portant sur la conception du module et de son articulation.

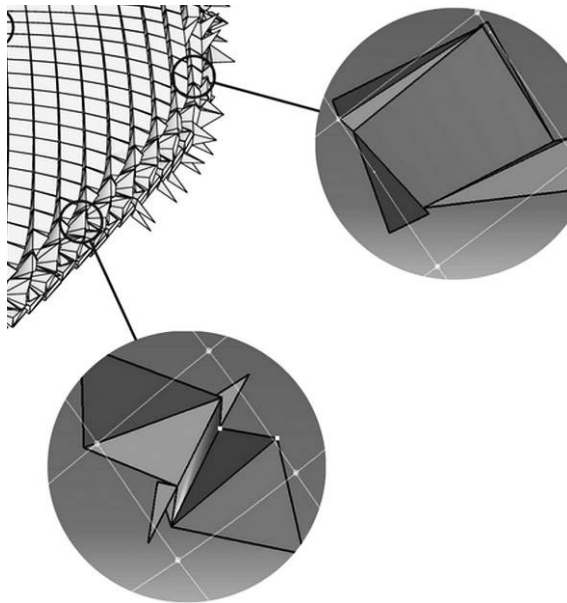


Figure 151.a Module de pliage

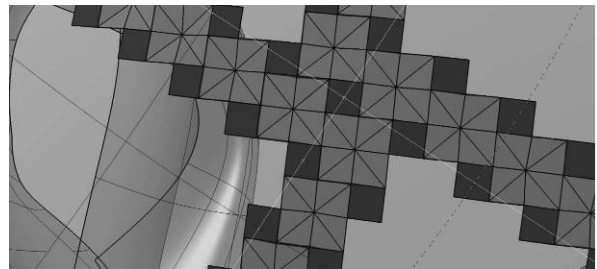


Figure 151.b Instanciation du module par discrétisation de la surface

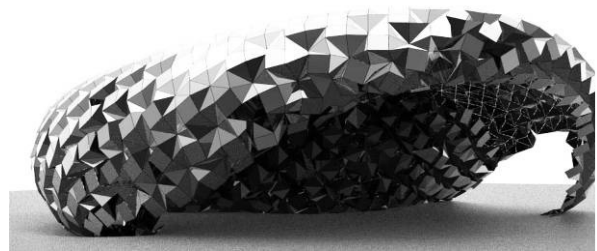



Figure 151.c Pavillon final

**Figure 151 : Images du projet "Folds", sources : Siim Tuksam**

La surface du pavillon résulte d’un test du module de pliage *via* une maquette physique (générée grâce à une machine de découpe laser). La faisabilité du pliage par son expérimentation a amené à valider le dessin de la surface du pavillon.

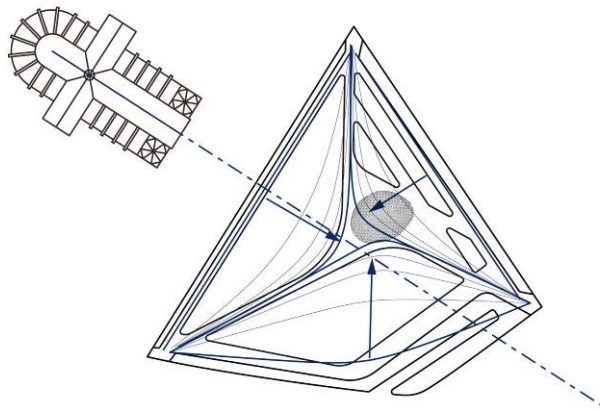
Lors de ce processus de conception, tout le projet a été représenté sur le modelleur Digital Project. Cependant, seule la géométrie des modules est paramétrée. Le lien des modules avec la surface de référence n’est pas paramétrique (cf. Figure 154).

Description des opérations cognitives impliquées et de leurs relations

| Opérations de Conception Architecturale  | Relation   | Opérations de modélisation paramétrique                      |
|--|--|--|
| Définition d’un module de « pliage » pouvant être répété sur une surface tout en restant constructible <i>via</i> une surface plane (cf. Figure 153)<br>(Echelle technique et technique) | <br>Opération de traduction géométrique | Conception d’une powercopy « module »<br>(Echelle technique) |

<sup>109</sup> Alors que la powercopy sur Digital Project est justement conçue pour pouvoir être liée à une référence. L’utilisation de la fonctionnalité « powercopy » ici va à l’encontre de l’intérêt que ces potentialités présentent. Ici la powercopy de l’étudiant semble avoir un défaut dans sa construction géométrique.

|   |                   |  |
|---|-------------------|--|
| Conception et positionnement d'une surface structurant l'espace du pavillon en fonction du flux du site (cf. Figure 152) (échelle fonctionnelle) et de la faisabilité technique du pliage (échelle technique) | →<br>Est suivi de | Représentation <i>a posteriori</i> de la surface |
|---|-------------------|--|

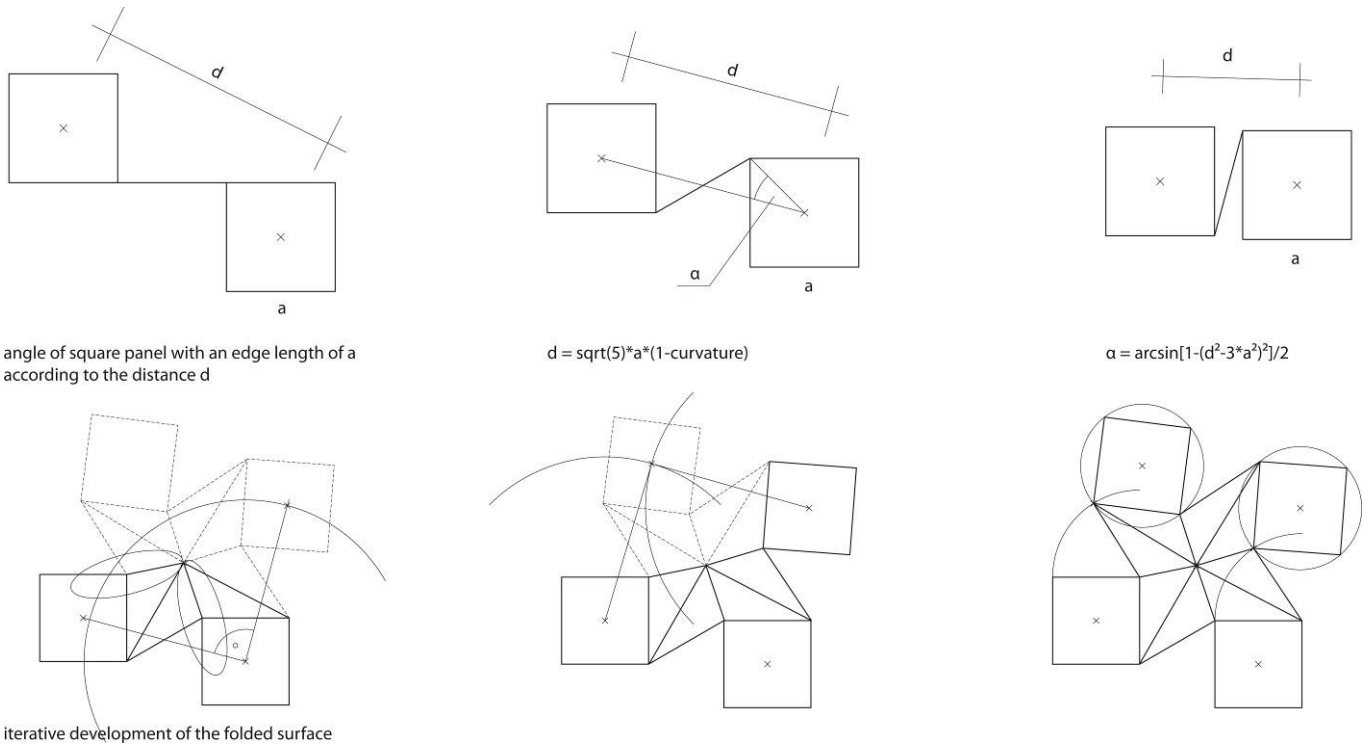


sidewalks are pushed to the center to concentrate the amount of people in the area.  
 acentral axis of the fotiv church is left open

**Figure 152: Positionnement du pavillon dans le site, source: Siim Tuksam**

Peut-on observer des « niveaux de conception » ?

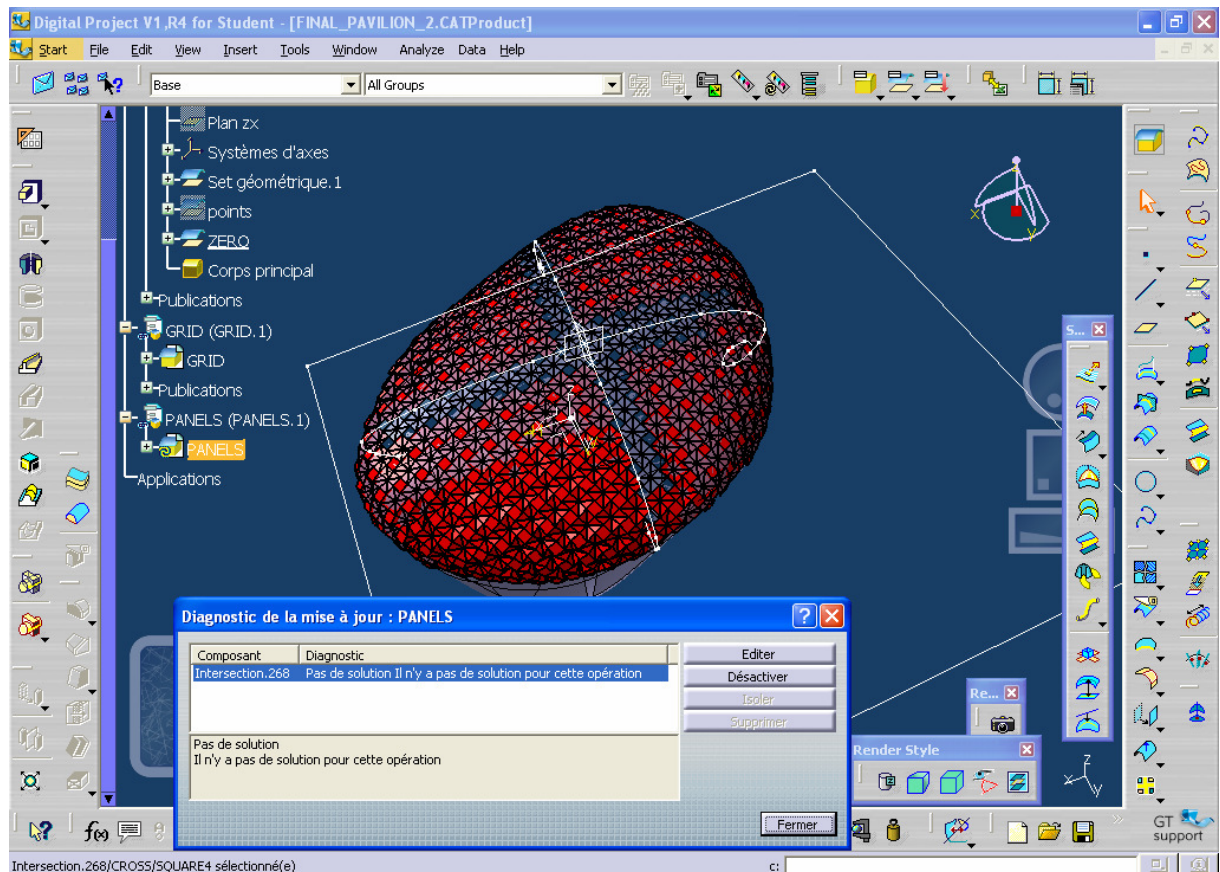
Un important travail de conception de la powercopy du module de pliage est observable, ceci s'inscrit dans un niveau de conception de modélisation paramétrique (cf. Figure 153). L'instanciation du module semble être ce qui a posé le plus de problèmes (techniques et géométriques) à l'étudiant (cf. Figure 154).



**Figure 153: Schéma explicitant la construction géométrique du module de pliage. Source : Siim Tuksam.**

Peut-on observer des opérations de modélisation paramétrique spécifiques ?

|                               |  |
|-------------------------------|--|
| Opérations de découpage       | Découpage du projet en une surface sur laquelle est instancié un module  |
| Opérations de dimensionnement | Mise en relation de la surface et du module <i>via</i> deux axes créés sur la surface (cette opération pose problème : une modification de la surface n'entraîne la mise à jour que des module situés sur ces axes cf. Figure 151 b) |



**Figure 154: Capture d'écran du modèle DP après modification de la surface définissant le pavillon, source: image d'A. de Boissieu à partir du modèle de S. Tuksaam.**

Peut-on observer des opérations logiques liées à la modélisation paramétrique ?

La difficulté de l'étudiant à pouvoir lier module de pliage et surface montre un manque de vérification du modèle paramétrique *via* des instances diverses.


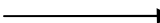
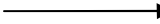

Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?

Données non disponibles.

## **2.7 Biocity, Olivier Scheffer (Mastere Spécialisé ENSCI)**

Description du projet et du modèle paramétrique développé

Le projet « biocity » vise à produire des gabarits enveloppe optimisés grâce à des algorithmes génétiques afin d'atteindre certains rendements énergétiques. La modélisation paramétrique a été faite en collaboration : O. Scheffer restant maître de son projet, A. de Boissieu a exploré et développé un modèle paramétrique permettant de répondre aux exigences du projet. La collaboration a été importante dans la conception du projet. Ainsi c'est *via* la prise en compte des capacités et exigences de Grasshopper et la discussion des intentions architecturales d'O. Scheffer qu'il a été décidé de : 1- partir d'une surface d'emprise correspondant à la parcelle (échelle parcellaire), pour chercher à optimiser 2- les hauteurs (coordonnées en Z des points du gabarit) et 3- la densité du gabarit enveloppe. Trois aspects étaient optimisés : une densité au sol visant à tendre vers 45% de surface bâtie, une prise au vent visant à être minimisée et des ombres portées en été visant à être le plus importants possible.

| Opérations de Conception Architecturale   | Relation  | Opérations de modélisation paramétrique   |
|---|---|---|
| Dimensionnement du gabarit enveloppe pour obtenir : - le plus d'ombre propre possible en été et le moins d'emprise au vent<br><br>(Echelle technique, économique et échelle géographique) | <br>Traduction en géométrie paramétrique   | Découpage du modèle : - en points dont la hauteur est variable, - et en sous surface pouvant être supprimées<br><br>Mise en relation de la surface du gabarit enveloppe avec un héliodon pour générer les ombres portées<br><br>(Echelle technique et géographique) |
|   | <br>Est suivi de                         | Modèle évalué et optimisé <i>via</i> galapagos et une fonction d'évaluation   |
|   | <br>Est suivi de                         | Développement d'un héliodon instancié dans le site<br><br>→ travail de modélisation qui a donné lieu à une méta modélisation, c'est-à-dire à une nouvelle modélisation en vue du partage <i>via</i> un format « sample »  |
| Partir de la parcelle totale<br><br>Echelle parcellaire   | <br>Traduction en géométrie paramétrique | Découpage du modèle en une surface d'emprise sur laquelle est générée une trame de points utilisée pour définir le gabarit enveloppe  |

Peut-on observer des « niveaux de conception » ?

On peut observer dans ce travail un niveau de modélisation relevant de la modélisation paramétrique, celui-ci correspond à la conception du modèle générant le gabarit enveloppe et faisant appel à des échelles géographique, technique, parcellaire et économique.

Un nouveau travail de modélisation paramétrique relevant d'un niveau de « méta modélisation paramétrique » visant à rendre le modèle de l'héliodon partageable et réutilisable par d'autres concepteurs.

Une recherche d'instances à été effectuée afin de démontrer les capacités du modèle (échelle optique) sur le cas de parcelles situées sur le plateau de Saclay.

Peut-on observer des opérations logiques liées à la modélisation paramétrique ?

Des tests du modèle paramétrique *via* des instances ont été faits régulièrement. Les résultats ont permis d'affiner le mécanisme du modèle paramétrique conçu (échelle technique ?) ainsi que le domaine de solutions défini.

Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?

**Opération pragmatique d'usage de l'outil :** Des opérations pragmatiques d'appropriation de l'outil ont été effectuées pour mettre en place de stratégies permettant d'optimiser les capacités de calcul de l'ordinateur. Des évaluations du temps de calcul permettaient fréquemment d'organiser le modèle ou les méthodes de travail pour atteindre un modèle réaliste et des calculs rapides.

**Opération pragmatiques de collaboration :** La collaboration entre OS et AB sur le modèle paramétrique a eu lieu alors qu'OS avait déjà élaboré ses intentions urbaines. Des discussions continues et une modélisation coopérative ont permis de mettre en place le modèle paramétrique. Des opérations pragmatiques d'interprétation ont permis à OS et AB de faire évoluer les pertinences et leur référentiel opératif commun en fonction des préoccupations de l'autre.

## **2.8 « L'objet de la conversation », Paola Assanti, Eunice Colella, Emilie Danel, Delphine Dargegen et Dani Jon (SP31)**

Description du projet et du modèle paramétrique développé

« L'objet de la conversation » est un projet où les étudiantes visaient à prendre en compte les « émotions » de visiteurs pour générer des formes. Ce projet se présente comme un fragment de processus. Il propose un modèle paramétrique Grasshopper qui permet de prendre en compte des échanges vocaux pour générer des géométries en prenant en entrée des ondes sonores (*cf.* Figure 155). Le but étant de permettre à des interlocuteurs de visualiser leur conversation et son évolution (*cf.* Figure 156).

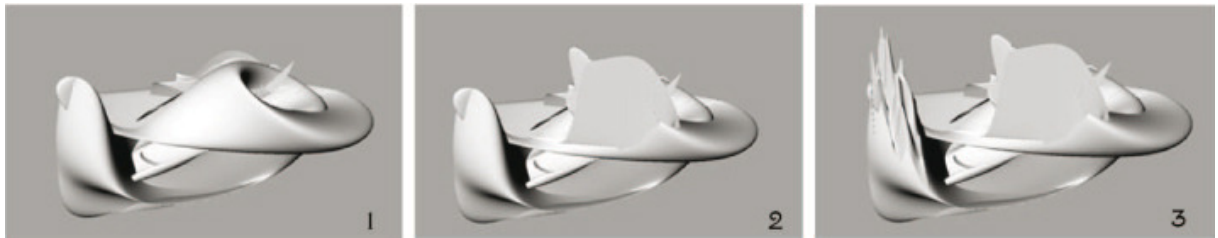


Figure 155: Evolution de la forme en fonction de l'avancée de la conversation, sources: P. Assanti, E. Colella, E. Danel, D. Dargegen et D. Jon

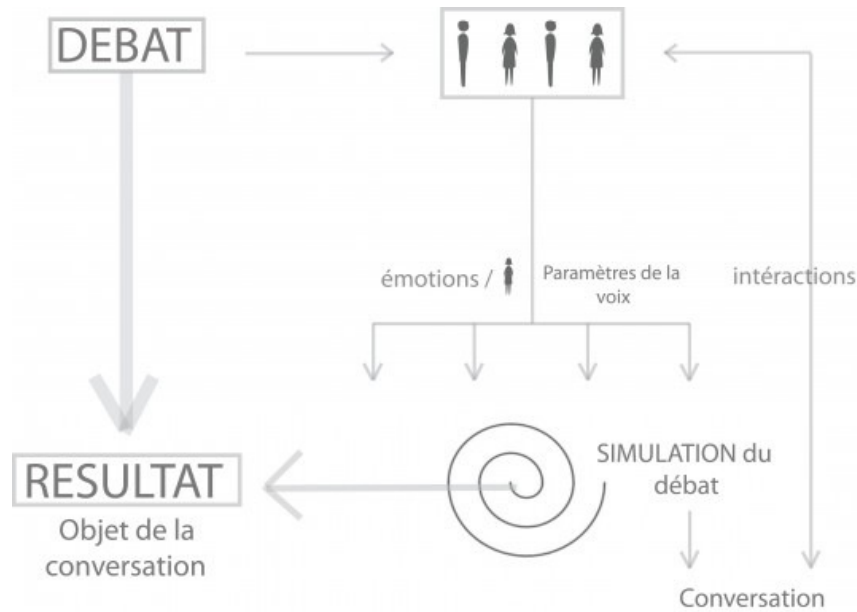
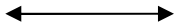



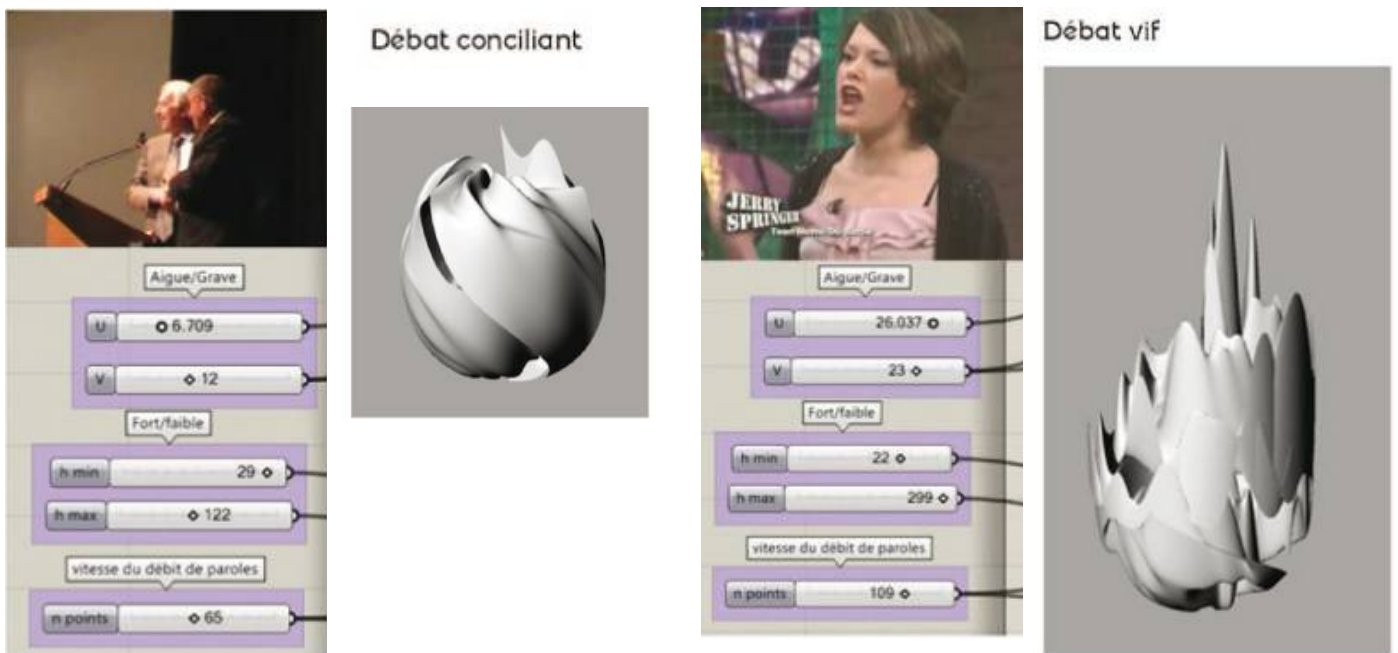
Figure 156: Schéma illustrant l'inscription du projet dans un processus de conversation, sources: P. Assanti, E. Colella, E. Danel, D. Dargegen et D. Jon

Pour concevoir leur modèle paramétrique les étudiantes ont réutilisé des codes disponibles sur internet ou fournis par les intervenants du cours, ceci pour 1- récupérer une variable numérique évoluant en temps réel en fonction des ondes sonores captées par un micro (*via processing*), 2- traduire ces variables en modification de géométrie (*via Grasshopper*), 3- figer sur Grasshopper ces géométries à intervalles réguliers pour enregistrer les évolutions de la conversation.

| Opérations de Conception Architecturale  | Relation  | Opérations de modélisation paramétrique   |
|--|---|---|
| Dimensionnement d'une géométrie en fonction des modulations de voix captées lors d'une conversation<br>Echelle géométrique ? | <br>Traduction en géométrie paramétrique<br>(par référence culturelle, imaginaire : le modèle est postérieur à l'idée) | Transformation de points d'une surface en fonction de variables numériques récupérés depuis processing <i>via GHowl</i><br><br>(récupération et appropriation de GHowl) |
| Dimensionnement de la géométrie en fonction d'un imaginaire de représentation  | <br>Est suivi de   | Transformation de points d'une surface en fonction de variables numériques définies   |



|  |   |   |
|--|---|---|
| des émotions (cf. Fantasia, référence amenée par les étudiantes)<br><br>Echelles de modèle et Socioculturelle  |   | manuellement : divisions en U et V en fonction de la tonalité aigue/grave de la conversation, altitudes min et max des points de la surface en fonction du niveau sonore, nombre de points définissant la surface en fonction de la vitesse du débit de parole (cf. Figure 157).<br><br>(variable devant être définie, méthode de définition interprétée et proposée pour validation) |
| Dimensionnement de la géométrie de l'objet pour rendre compte des évolutions de la conversation dans le temps. | <p style="text-align: center;">↔</p> <p style="text-align: center;">Traduction en géométrie paramétrique</p> <p style="text-align: center;">(référence aux capacités de GH pour la conception architecturale, puis inférence sur le modèle)</p> | Utilisation d'un sample trouvé sur internet pour « cuire » les surfaces produites à intervalle régulier   |



**Figure 157:** schéma illustrant l'instanciation manuelle de certaines des variables définissant la géométrie de "l'objet de la conversation, sources: P. Assanti, E. Colella, E. Danel, D. Dargegen et D. Jon

Peut-on observer des « niveaux de conception » ?

Le travail des étudiantes a surtout porté sur l'instanciation, puisque l'activité de réappropriation (correspondant au niveau de conception du modèle paramétrique) de samples (relevant de la méta-modélisation paramétrique) a surtout été faite par l'intervenant (AB).

Peut-on observer des opérations logiques liées à la modélisation paramétrique ?

L'exploration des instances a été une forme d'opération de vérification.

Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?

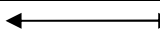
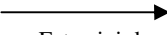
Ce travail s'est beaucoup appuyé sur la collaboration avec les intervenants du cours (pour le lien avec Processing et Grasshopper lui même) comme avec la communauté internet, le modèle reprenant trois samples différents.

## 2.9 « Morphoclics » de Anne-Laure Farcy, John Carvalho, Mick Ben Susan, Julie Mistoco et Annael Médina (SP31)

Description du projet et du modèle paramétrique développé

« Morphoclics » est un projet visant à rendre compte l'« activité inconsciente » des utilisateurs d'une salle informatique de l'ENSA-v, cette « activité inconsciente » étant les gestes d'utilisation de l'ordinateur : cliquer, déplacer une souris, taper sur un clavier, *etc.* Ces gestes, les étudiants les reprennent en les disjoignant de leurs fonctions premières (utiliser une application par exemple) et les utilisent comme facteur de génération d'une forme. Cette forme résulte en fait de la déformation d'un objet « modèle de l'école » en fonction de variables numériques générées par l'activité de la salle info.

Pour ce modèle, les nombres de clics et de scroll sont captés grâce à un programme puis envoyé par internet, où ces valeurs sont traitées par Processing et envoyées à Grasshopper. Le modèle grasshopper déforme un volume « modèle de l'école » grâce à un « morphing ». La technologie « morphing » vise à adapter le volume à une « boîte cible », celle-ci pouvant être déformée facilement.

| Opérations de Conception Architecturale   | Relation   | Opérations de modélisation paramétrique  |
|---|--|--|
| Dimensionnement d'une géométrie en fonction des activités d'utilisation d'une salle informatique<br>Echelle ?     | <br>Traduction en géométrie paramétrique (par référence culturelle, imaginaire : le modèle est postérieur à l'idée) | Transformation d'un volume en utilisant une technique de morphing, qui consiste à modifier un « boîte cible » dans laquelle sera contraint le volume déformé. Les « clics » récupérés <i>via</i> processing sont utilisé comme variable pour modifier les points définissant cette « boîte cible ».<br><br>( récupération et appropriation du sample Morphing) |
| Définition des éléments définissant la dimension de l'objet transformé : nombre de clics, fréquences, <i>etc.</i> | <br>Est suivi de  | Définition des « inputs »  |

Peut-on observer des « niveaux de conception » ?

Comme pour le projet « l'objet de la conversation », le travail de ces étudiants a surtout impliqué l'appropriation d'un modèle développé par quelqu'un d'extérieur. Le travail relève donc, avant tout, de la modélisation paramétrique grâce à la collaboration et de l'instanciation.

Peut-on observer des opérations logiques liées à la modélisation paramétrique ?

L'exploration d'instances a été immédiate et a contribué à l'élaboration du modèle paramétrique. Les étudiants n'avaient pas d'idées précises des instances visées : ils n'ont donc pas eu recours à la « généralisation ».

Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?

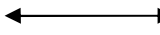

Ce travail s'est beaucoup basé sur la collaboration avec un intervenant expert en modélisation paramétrique. La collaboration a amené à affiner les intentions des étudiants et leur traduction en mécanismes paramétriques. C'est l'intervenant qui a proposé le recours à la technique du « morphing » à partir du discours des étudiants. Après une première proposition de modèle et une description des mécanismes, les étudiants se sont appropriés par eux-mêmes le mécanisme de déformation.


## **2.10 « Danse-city » de Marie Brossard, Alice Huet, Camille von KNECTEN, Elsa Favier, Marion Gomez et Caroline Michel**

Description du projet et du modèle paramétrique développé

« Danse-city » est un projet d'installation visant à faire interagir des spectateurs avec une génération de formes. Des tapis au sol sont munis de capteurs provoquant des transformations de géométries sur Grasshopper. Les transformations ont lieu en temps réel et sont projetées sur un écran visible depuis l'espace des capteurs.

L'analyse de ce projet n'est pas aisée puisque l'objet de la conception de ce projet est l'installation, et le modèle paramétrique est une partie de cette installation.

| Opérations de Conception Architecturale                        | Relation  | Opérations de modélisation paramétrique  |
|--|---|--|
| Exigence de lisibilité de l'installation<br>Echelle optique    | <br>Traduction en géométrie paramétrique | Opération de découpage : le modèle se base sur un « cube de départ », auquel on fait subir des transformations.<br><br>Echelle géométrique |
| Définition des modalités de transformation d'un cube de départ |    | Opération de mise en relation d'un « cube de départ » avec des opérations booléennes simples proposées par Grasshopper                     |

|   |   |  |
|---|---|--|
| Echelle géométrique   | Mise en commun  | Echelle technique (lié aux fonctionnalités de Grasshopper) et géométrique  |
| Exigence d'interactivité de l'installation<br>Echelle fonctionnelle | <br>Mise en commun<br>Choix d'interaction lié à la culture du paramétrique et du numérique | Mise en relation de la génération de géométrie avec des variables indépendantes générées par les captations par Arduino des mouvements des spectateurs |

Peut-on observer des « niveaux de conception » ?

Le travail a porté sur une définition d'un modèle paramétrique, construit grâce aux échanges entre étudiants et intervenant, cela relève d'un niveau de conception relatif à la modélisation paramétrique. L'exploration d'instances a lieu par les spectateurs sous forme de jeu ou de danse. Si le domaine de solutions est maîtrisé par les concepteurs, l'exploration des instances se fait de façon libre.

Peut-on observer des opérations logiques liées à la modélisation paramétrique ?

Données non disponibles.

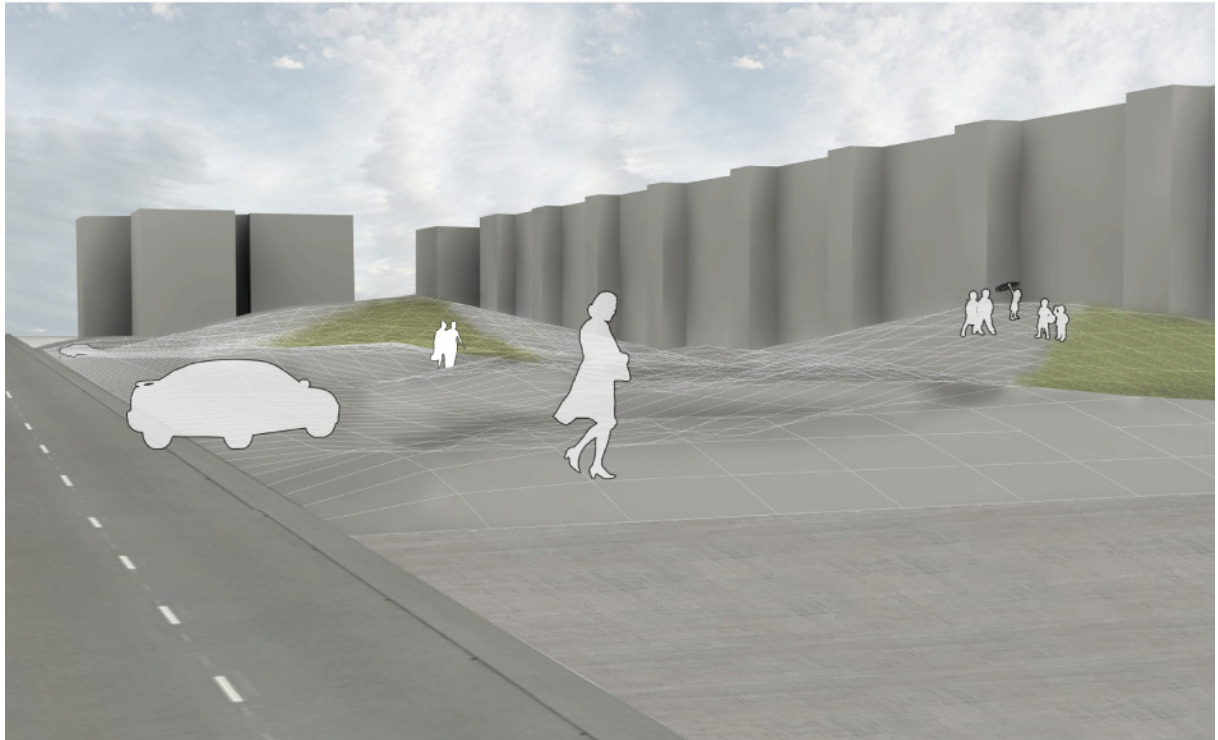
Peut-on observer des opérations de collaboration liées à la modélisation paramétrique ?

Ce travail s'est beaucoup basé sur la collaboration avec un intervenant expert en modélisation paramétrique.

## **2.11 « Topographies », Aurea Rodriguez, Pablo Gancedo, Samya Pelloquin et Mathias Saboya (Architecture Paramétrique)**

Description du projet et du modèle paramétrique développé

Le projet « Topographies » (cf. Figure 68) vise à aborder l'espace d'un des parkings de la « cité des 4 000 » (Paris), comme nouveau paysage et espace de transition dans le parcours effectué par les habitants de la cité entre le moment où ils sortent de leur voiture et celui où ils entrent dans leur immeuble. Différents usages devaient être gérés par cette surface.



**Figure 158 : Extrait de la présentation du projet Topographies d'Aurea Rodriguez, Pablo Gancedo, Samya Pelloquin et Mathias Saboya**



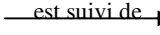
L'intention des étudiants était de développer la modélisation d'une surface pour générer une « topographie » (échelle sémantique). Après discussion avec les étudiants (opération pragmatique d'interprétation), je leur ai proposé de concevoir le modèle paramétrique à partir de l'emprise du parking (échelle parcellaire) et de faire réagir cette surface à des objets géométriques, comme le fait le sample existant : créer\_srf\_repulseur (échelle de modèle). En effet l'intention était de créer une surface réagissant de façon « topologique »<sup>110</sup> aux contraintes de cheminement (échelle fonctionnelle).

Le modèle possède une emprise au sol correspondant au parking (op. de découpage associée à l'échelle parcellaire) utilisée une version réappropriée du sample créer\_srf\_repulseur (échelle de modèle) pour faire réagir les hauteurs de la surface non pas à des points mais à des courbes représentant les cheminements (échelle fonctionnelle).

L'exploration des instances a vite montrée que la hauteur maximale et le rapport au sol de la surface générée n'étaient pas contrôlables. Cela a posé un problème dans la relation entre la surface générée et l'immeuble adjacent (échelle de voisinage) parce que la surface risquait de boucher les vues des habitants du 1<sup>er</sup> étage. Le problème venait de la création des hauteurs de la surface : celles-ci étaient directement générées par les distances aux courbes attracteurs (c'est-à-dire : la parcelle de base est divisée en n points, on calcule pour chacun des points sa distance à la « courbe attracteur » la plus proche, cette distance est utilisée pour donner une hauteur à la surface à la verticale de chaque point). Les étudiants ont décidé de « mapper » la liste de distance pour la transformer en liste de hauteurs contrôlables. La fonction « map » est proposée par Grasshopper (échelle de modèle). Elle permet de garder les mêmes proportions entre les valeurs, mais de les répartir dans un domaine maîtrisé, ici

<sup>110</sup> Entretien avec les étudiants, le 11 juin à l'ENSA de Marnes la Vallée.

défini en fonction du rapport au sol (échelle fonctionnelle) et à la hauteur maximale voulus (ech. de voisinage et de visibilité).

| Opérations de Conception Architecturale   | Relation  | Opérations de modélisation paramétrique  |
|---|---|--|
| Découpage de la dimension d'emprise du projet, celle-ci correspondant au parking.<br>Echelle parcellaire  | <br>Traduction en géométrie paramétrique | Découpage du modèle pour partir d'une surface d'emprise correspondant au parking.<br>(Opération de modélisation assistée par des opérations de collaboration ) |
| Volonté de faire référence à une « topographie» (échelle de modèle et sémantique)<br><br>Volonté de faire réagir la surface « topographique» aux cheminements et aux usages voulus (échelle fonctionnelle)  | <br>Traduction en géométrie paramétrique | Reprise du sample créer_srf_reuplreur<br>(Opération de modélisation assistée par des opérations de collaboration )   |
| Volonté de contrôler les hauteurs max pour un effet séduisant (éch. optique) ainsi qu'un rapport acceptable au vis-à-vis généré par les logements (Echelle de voisinage).<br><br>Volonté de contrôler les hauteurs min pour le rapport au sol (Echelle fonctionnelle) | <br>est suivi de                        | Adaptation du sample créer_srf_reuplreur pour pouvoir contrôler les altitudes max et min de la surface créée (mapping).  |

Peut-on observer des « niveaux de conception » ?

Il semble que ce travail s'appuie surtout sur une appropriation d'un code existant et une instanciation de celui-ci en fonction des échelles architecturales visées. Ce travail, très proche de l'instanciation plutôt que de la recherche de système, s'est pourtant révélé très riche et a même amené à la production d'un nouveau sample (mapping list).

Peut-on observer des opérations de modélisation paramétrique spécifiques ?

|                            |  |
|----------------------------|--|
| Opérations de découpage    | Découpage de la surface d'emprise  |
| Opérations dimensionnement | Mise en relation des altitudes de la surface avec des objets spécifiques : des points dans le sample original, modifié en polygones par les étudiants. |

Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?

C'est au travers d'une opération de vérification du modèle *via* l'instanciation que le problème du contrôle des hauteurs est apparu.

Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?

Ce travail a procédé d'allers-retours fréquents avec un expert en modélisation (les étudiants n'étant alors qu'au début de leur apprentissage). Ceci est passé par des opérations d'interprétation. Il leur a été proposé le recours à un modèle déjà développé qu'ils ont pu par la suite s'approprier (le sample créer\_srf\_repulseur).

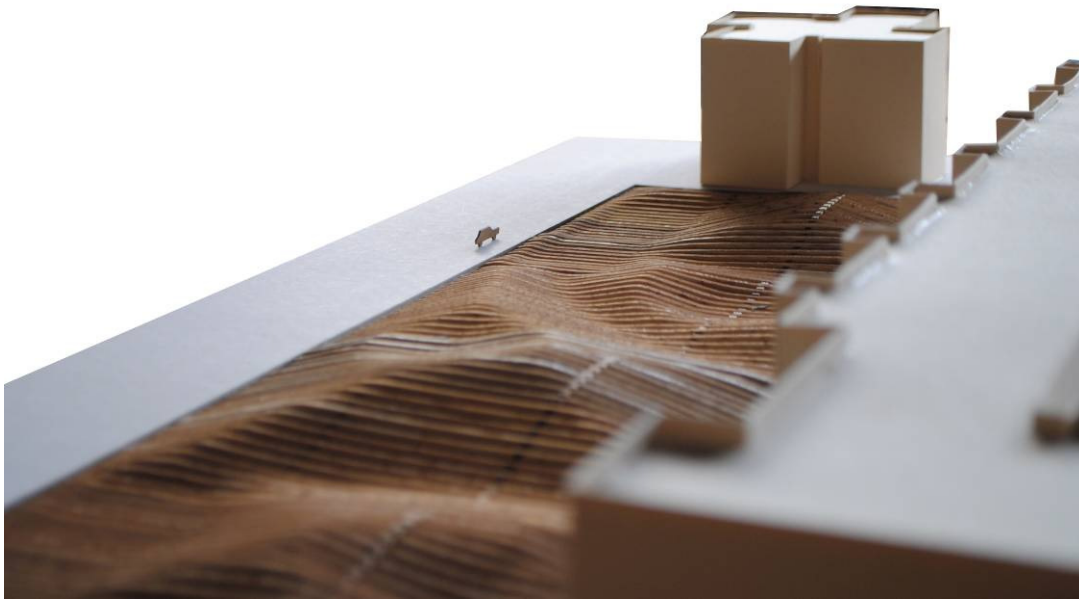


Figure 159: Photo de la maquette "Topographies", sources: Aurea Rodriguez, Pablo Gancedo, Samya Pelloquin et Mathias Saboya

## Analyse de pratiques professionnelles

### ***2.12 Pavillon de la fondation Louis Vuitton pour la Création, Gery Technologies (Paris)***

Description du projet et du modèle paramétrique développé

La modélisation paramétrique a été utilisée lors de la conception du Pavillon de la Fondation Louis Vuitton pour la Création. Les premières esquisses du bâtiment pensé par Gehry Partners datent de 2006. Sa construction a commencé en 2009 au Jardin d'Acclimatation à Paris. Pour sa conception,

plus de trois cents professionnels aux compétences différentes et géographiquement très dispersés, sont simultanément intervenus. A l'occasion de ce projet, Gehry Technologies, société de développement de Digital Project, a fondé le bureau Gehry Technologies Europ111 à Paris en 2007, afin de former les intervenants du Pavillon Louis Vuitton à Digital Project. Elle les a également assistés dans la conception du modèle paramétrique commun (Witt, 2009).

#### Observation spécifique d'occurrences d'opérations de modélisation paramétrique de découpage

Dans le modèle paramétrique développé par les bureaux d'études rassemblés à Paris on observe deux découpages différents qui correspondent à des temps différents : ce sont le découpage par lots et le découpage par « nœuds ».

- Découpage par lots. Ce découpage correspond à la répartition des ouvrages entre les différentes équipes. Il est lié à l'organisation traditionnelle du chantier. On distingue ainsi le lot « architecture et finition » qui appartient à l'architecte local de l'agence Studios Architecture, le lot « gros œuvre » de SETEC ou encore, le lot « parois vitrées » de RFR, *etc.* Ce découpage répond à une pertinence liée à l'organisation du travail. Il permet d'identifier les objets géométriques sur lesquels chaque équipe travaille spécifiquement. Ce découpage est associé à des droits d'écriture qui dépendent de la hiérarchie de l'intervenant. Ce découpage de la conception crée un cloisonnement entre les sous-modèles sur lesquels travaillent les différentes équipes.

Si ce découpage semble avant tout fonctionnel, il répond également à des distinctions opérées par Gehry Partners. Dans la « géométrie morte » transmise à Paris, on retrouve un système de nominations qui reflète des intentions de conception. Ces nominations distinguent des éléments contrastés : ainsi « les voiles » désignent une partie vitrée faisant signal, et l'« iceberg » un « bâtiment dans le bâtiment » (Gehry, 2010) pour les espaces d'exposition. Ces partis architecturaux sont lisibles dans le choix des matériaux et du traitement des éléments (vitrage, béton, *etc.*) et, se retrouvent, de fait, dans le découpage des ouvrages... Ainsi le découpage de l'« iceberg » correspond au gros œuvre et celui des « voiles » aux verrières.

- Découpage par « nœuds ». Ce découpage porte sur les zones où la rencontre de lots est particulièrement complexe. Contrairement au découpage précédent qui était composé de « products » et qui correspondait à la structure même du modèle, ce découpage est surajouté: il se fait par script et crée un nouveau « product ». Ces découpages, appelés « section box » par Gehry Technologies (de Boissieu, 2009), sont au nombre de quatre-vingt environ. Ils sont mis en œuvre lors de réunions régulières menées par Gehry Technologies et rassemblent les équipes concernées par les ouvrages en cause. Ces réunions visent à vérifier la cohérence du modèle et à coordonner les zones sensibles. Cela permet par exemple de contrôler et de partager les informations entre le travail de Setec et celui de RFR, sur les points d'appuis de la structure des verrières sur les ouvrages en béton. Ces reprises de charges sont gérées par des éléments architecturaux spécifiques: les « tripodes ». La modification d'un tripode par l'équipe chargée des verrières (RFR) doit immédiatement être transmise à l'équipe qui dessine et calcule le gros œuvre (Setec). Tandis que le précédent découpage cloisonnait les objets de chaque équipe, ce second découpage permet un contrôle de la cohérence du travail entre les équipes et un partage des informations.

---

<sup>111</sup> [www.gehrytechnologies.com/](http://www.gehrytechnologies.com/)



Ces deux découpages du modèle sont donc liés à l'organisation du travail et à la répartition des tâches entre les différents intervenants. A ce niveau général d'observation, le modèle semble avoir été construit pour optimiser la communication et le partage du travail post-conception.

Du « découpage par lots », nous avons choisi les lots « parois vitrées » et « architecture et finitions » pour approfondir nos analyses.

Le modèle « parois vitrées », imparti à RFR, concerne les parois du bar, des lobbies, des circulations verticales, du forum et des tours. Ce modèle est constitué d'un sous-modèle par paroi. Pour chacun de ces éléments on observe une même organisation. Ces sous-modèles se constituent respectivement de six parts, distinguant les différents éléments de structure (fig.3).

| Part                      | Eléments représentés  |
|---------------------------|---|
| _ Verre                   | Surface du verre  |
| _ Structure<br>Tertiaire  | Structure du verre  |
| _ Structure<br>Secondaire | Support de la structure tertiaire   |
| _ Tripode                 | Structure reliant la structure secondaire à la structure primaire (le gros œuvre)   |
| _ Surface de<br>référence | « Part » à partir de laquelle se construisent les géométries. Elle est établie en référence directe avec le modèle des architectes à partir d'une copie de l'élément. |
| _ Publications            | Informations sur les points partagés avec d'autres « Products »   |

**Figure 160 : Arbre des spécifications des modèles des parois vitrées**

Le découpage par structure primaire, secondaire et tertiaire correspond à une organisation courante du travail des BET structures<sup>112</sup>. Cette organisation permet de hiérarchiser le rôle et les performances des différents éléments de structure. A ces éléments de structure s'ajoutent la géométrie produite par Gehry Partners : elle sert de surface de référence pour la construction de la géométrie du nouveau modèle. De même, est ajouté un ensemble de données partagées avec les autres sous-modèles (des altimétries, des points de références partagés, *etc.*). Ces données prennent la forme de « publications ». Le terme « publication » est propre au vocabulaire de CATIA. Les « publications » sont des données validées par le concepteur comme « partageables » par d'autres intervenants, c'est-à-dire suffisamment stables et valides pour être le support de travail d'autres concepteurs.

On observe ici que les opérations de modélisation de découpage du modèle « parois vitrées » assistent à la fois des opérations de conception architecturale et la collaboration entre les acteurs. Les opérations de la conception architecturale assistée par Digital Project sont celles portant sur la structure. Le découpage du modèle en fonction de son rôle mécanique en structures tertiaires et secondaires répond à une opération de conception relevant d'une pertinence technique. La collaboration entre les acteurs est accompagnée à travers l'usage des « surfaces de références » et des « publications ». La « publication » de données permet d'identifier les éléments communs entre les sous-modèles, par exemple pour clarifier le point pertinent à partager entre le « gros œuvre » et le lot « parois vitrées » pour ancrer un tripode. La « publication » et l'utilisation de « surface de référence »

<sup>112</sup> Le sous-modèle du lot « verrières » est découpé de la même façon.

clarifient le choix des données partagées et permettent ensuite de mettre à jour la cohérence des sous-modèles entre eux. Ici le « paramétrique » joue un rôle important dans la collaboration entre les intervenants.

Les finitions et l'aménagement des espaces intérieurs et extérieurs sont impartis au lot de l'architecte local de l'agence Studios Architecture. Ce sous-modèle est assez complexe car il concerne des espaces fragmentés et sans identités de niveaux ou de fonctions. Le modèle s'organise en découpages successifs (fig. 4) et permet un découpage systématique qui régit toutes les zones du sous-modèle.

|  |
|--|
| <p>1 découpage du modèle par sous-modèles :</p> <ul style="list-style-type: none"> <li>_ Ouest</li> <li>_ Centre</li> <li>_ Est</li> <li>_ Sous-sol</li> </ul>   |
| <p>2 découpage des sous-modèles par ouvrages :</p> <ul style="list-style-type: none"> <li>_ Niveaux</li> <li>_ Circulations</li> <li>_ Façades</li> </ul>  |
| <p>3 découpages des ouvrages pour la construction géométrique :</p> <ul style="list-style-type: none"> <li>_ « Géometric set » (objets géométriques)</li> <li>_ Paramètres (épaisseurs, <i>etc.</i>)</li> <li>_ « Publications » (données communes)</li> </ul> |

**Figure 161 : Découpage du modèle « architecture et finitions »**

Un premier découpage reprend une distinction donnée dès le début du projet par Gehry Partners. Il identifie quatre zones : « ouest », « est », « centre » et « sous-sol ». Ce découpage a une pertinence principalement géographique (comme l'indiquent les noms « est », « ouest » *etc.*), mais il est aussi fonctionnel et technique. En effet, il correspond à un groupement des ouvrages par zones indépendantes qui lui donnent une pertinence fonctionnelle. La zone « centre » est séparée de la zone « est » par un joint de dilatation (pertinence technique).

Chacune des zones « ouest », « est », « centre » et « sous-sol » est ensuite découpée par ouvrages en distinguant les niveaux, les noyaux de circulation et les façades. Distinguer les niveaux n'aurait pas eu de sens pour découper la globalité du modèle à cause de la non-coïncidence des altimétries. Distinguer les façades est nécessaire car celles-ci sont régulièrement modifiées par les différents bureaux d'études qui les gèrent. Dans ce dernier découpage les ouvrages (les noyaux, les façades, *etc.*) sont identifiés en fonction de l'organisation du travail de l'entreprise générale. Ce découpage vise explicitement à favoriser la cohérence de la communication entre l'agence Studios Architecture et l'entreprise générale. Le découpage du sous-modèle « architecture et finitions » vise donc à assister l'organisation du travail plus que la conception du bâtiment lui-même.

Un dernier découpage distingue ensuite trois éléments dans chacun des ouvrages : - un « geometric set » qui regroupe les informations géométriques sur les parois, les sols, les plafonds, les portes, *etc.*, - des « paramètres » qui définissent par exemple l'épaisseur standard des faux plafonds, les décalages des sols en bois ou en résine, *etc.* et – des données communes, sous la forme de

« publications », telles que les altimétries, les points partagés par plusieurs ouvrages, *etc.* La construction géométrique est donc paramétrée à travers un certain nombre de variables et de références extérieures.

Les premiers découpages du modèle du lot « architecture et finitions » (par sous-modèles, puis par ouvrages) correspondent à des opérations de modélisation liées à l'organisation du travail et, visent à accompagner la collaboration entre intervenants, en particulier entre l'architecte et l'entreprise générale. Le dernier découpage (en « geometric set », « paramètres » et « publications ») assiste, quant à lui, des opérations de conception de l'architecture elle-même, portant, entre autres, sur l'organisation de l'espace, la caractérisation des parois ou l'accessibilité handicapée.

### **2.13 « Babel » Milovan Yanatchkov, RVBA (Paris 2010)**

Cette analyse a une forme différente : nous avons gardé ici sa rédaction originale, formalisée en vue d'une publication dans la revue d'art plastique « Contraintes ».

r,v,b,a est un atelier qui permet la rencontre entre architectes, hackers et artistes. Le terme « hacker » possède plusieurs définitions. Nous ne reprenons pas ici l'acception la plus couramment utilisée dans les médias de « pirates » ou « crackeur », mais celle plus ouverte de développeurs non professionnels. A l'origine hacker signifie « bidouilleurs ». Dans cet atelier, Milovann Yanatchkov travaille la programmation, le code, comme une « matière » pour penser l'architecture. Cette image de « matière » est celle qu'emploie Milovann Yanatchkov. Il dit « travailler le code informatique comme [il] le ferait de terre glaise. » [Yanatchkov, 2010]. Certaines de ses expérimentations architecturales s'appuient sur le logiciel (fluxus).

(fluxus) propose un environnement de programmation pour coder des formes en Open GL, c'est-à-dire en lien direct avec les bibliothèques graphiques fonctionnant sur les systèmes d'exploitation de l'ordinateur. (fluxus) est l'œuvre d'un artiste hacker du collectif PAWFAL (Poor Artists Working For A Living [pawfal.org]), Dave Griffiths. (fluxus) possède une interface réduite au minimum : c'est un simple éditeur de texte. Il ne comprend ni boutons, ni boîtes de dialogues : on est face au code seulement et à ce qu'il produit. (fluxus) se compose d'une fenêtre entièrement noire, animée d'un seul curseur, et celle-ci s'emplit au fur et à mesure que l'artiste tape le code. À n'importe quel moment la commande <ctrl+E> fait « s'exécuter » le texte (*cf.* Figure 162). Performativité des mots qui deviennent formes. Le texte s'exécute et l'écran s'anime du projet. On se retrouve devant la présence simultanée du texte et de la forme. Le code est alors augmenté de son interprétation formelle.

Ce texte est modifiable à chaque instant : l'artiste peut « revenir dans le passé » pour modifier une variable ou une commande, et la forme évoluera. Comme l'écrit Dave Griffiths, (fluxus) est « *une succession continue de changements* » (« *A continuous succession of changes* » : cette continuité est dans la succession d'opérations et non dans le système qui, lui, est non continu et discret.) [pawfal.org/fluxus/].

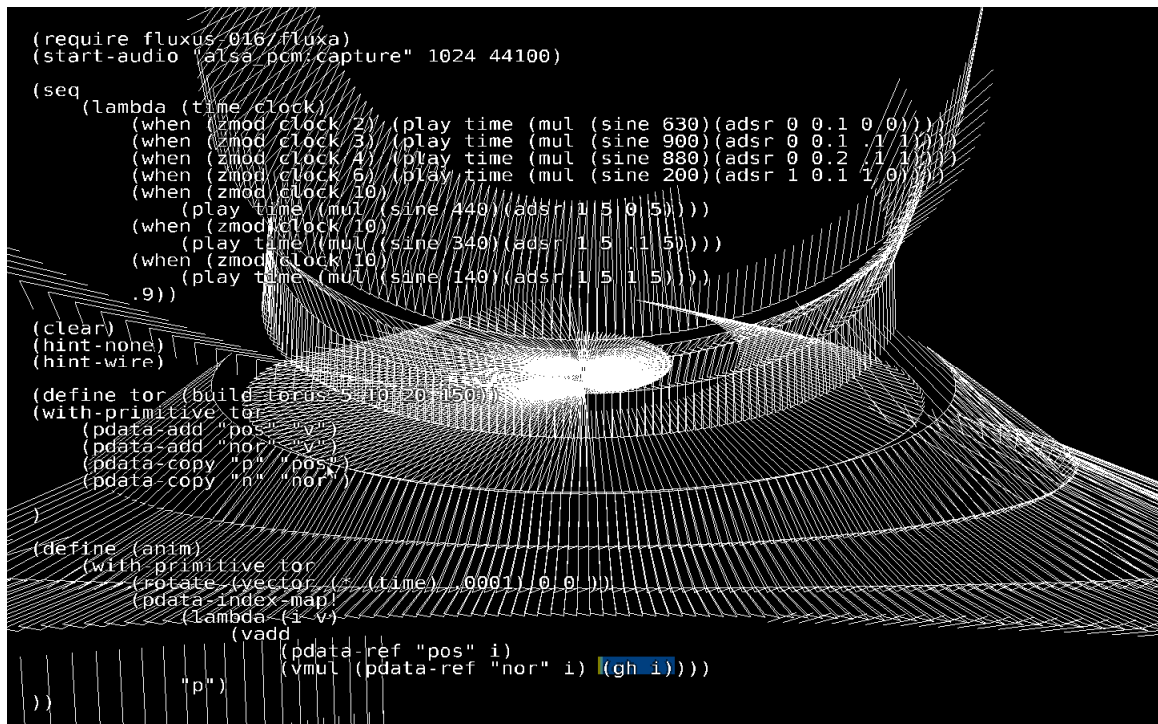


Figure 162 : « Instantané » d'une expérimentation de r,v,b,a sur (fluxus) sur lequel on lit la juxtaposition du code et de la forme

(fluxus) a initialement été développé pour concevoir rapidement des jeux et pour l'apprentissage de la programmation graphique en 3D. Il a ensuite été utilisé pour des performances de « livecoding » [pawfal.org] [toplap.org] pendant lesquelles l'artiste programme en temps réel, pour provoquer des sons et des images. Que (fluxus) ait été utilisé pour des performances puis approprié en architecture est significatif. Ce déplacement peut être rapproché des propos d'Antoine Picon : « [la] commensurabilité entre forme et événement constitue à notre sens le fondement de la tendance que l'on observe aujourd'hui de considérer l'architecture comme performance » [Picon, 2010. p. 75].

### Processus et évènements

(fluxus) s'inscrit dans des expérimentations (cf. [rvba.fr] [pawfal.org] [taplop.org]) dont l'objet est le processus et non plus la perception finale de l'œuvre. Ce déplacement est radicalisé dans le cas du livecoding : le processus figuré dans la programmation devient l'œuvre d'art et se donne à voir en tant que tel. L'œuvre est alors un flux, à la fois continue et non-achevée. Elle est toujours modifiable et les formes qu'elle génère peuvent être explorées sans fin. Elle constitue un flux modulé par des variations dont des pics d'intensités surgissent parfois. Ces pics sont des « évènements », des formes singulières. Ils sont des accidents, des « objets-trouvés » (Yanatchkov, 2010), et n'existent que comme moments figés du processus. Au sens propre comme au figuré, sur (fluxus) ces évènements sont des « instantanés ».

Le mot même « processus » possède une polysémie révélatrice. Ce terme permet à la fois de désigner l'activité de l'artiste lorsqu'il conçoit son œuvre, c'est la définition utilisée dans cet article, mais il désigne aussi, dans le vocabulaire technique lié à l'informatique, l'exécution d'instructions par un ordinateur : c'est la définition utilisée par Milovann Yanatchkov lorsqu'il parle de son travail

[Yanatchkov, 2010]. Dans le cas du livecoding, l'affirmation: « *l'objet de la conception est le processus* » peut être comprise littéralement.

Milovann Yanatchkov utilise (fluxus) pour questionner et expérimenter la conception architecturale. Un des aspects de son travail interroge spécifiquement l'usage de la programmation dans une expérimentation de formes autour de l'idée de « tour », pour le concours « *evolo tower* » de 2009.

Comment se déroule le processus? Dans un premier temps, M. Yanatchkov code les caractéristiques qu'il veut donner à sa tour. Des dalles qui peuvent pivoter, se déformer et se déployer. Des hauteurs variables, des possibilités de réseaux de tours. Des surfaces-façades entre les dalles, rythmées ou striées, accompagnées de transparences. Puis, sont associés à ces descriptions des facteurs aléatoires et des possibilités de déformation.

Ensuite, à partir de ce code, une infinité d'exécutions peut être lancée. En spécifiant des hauteurs différentes, en décrivant une tour ou un réseau de tours, avec dix, puis cent étages. Toutes les formes sont différentes. M. Yanatchkov explore le flux du processus qu'il a mis en place au travers de ces variations et de ces intensités (cf. Figure 163).

À l'écriture attentive et méticuleuse du code se substitue le lâcher-prise de l'exploration : « *Ces formes, c'est comme si ça n'était pas moi qui les avaient faites* » [Yanatchkov, 2010]. La posture de l'artiste se transforme. Il conçoit le code, mais souhaite ne pas contrôler pas les formes qui émergent.

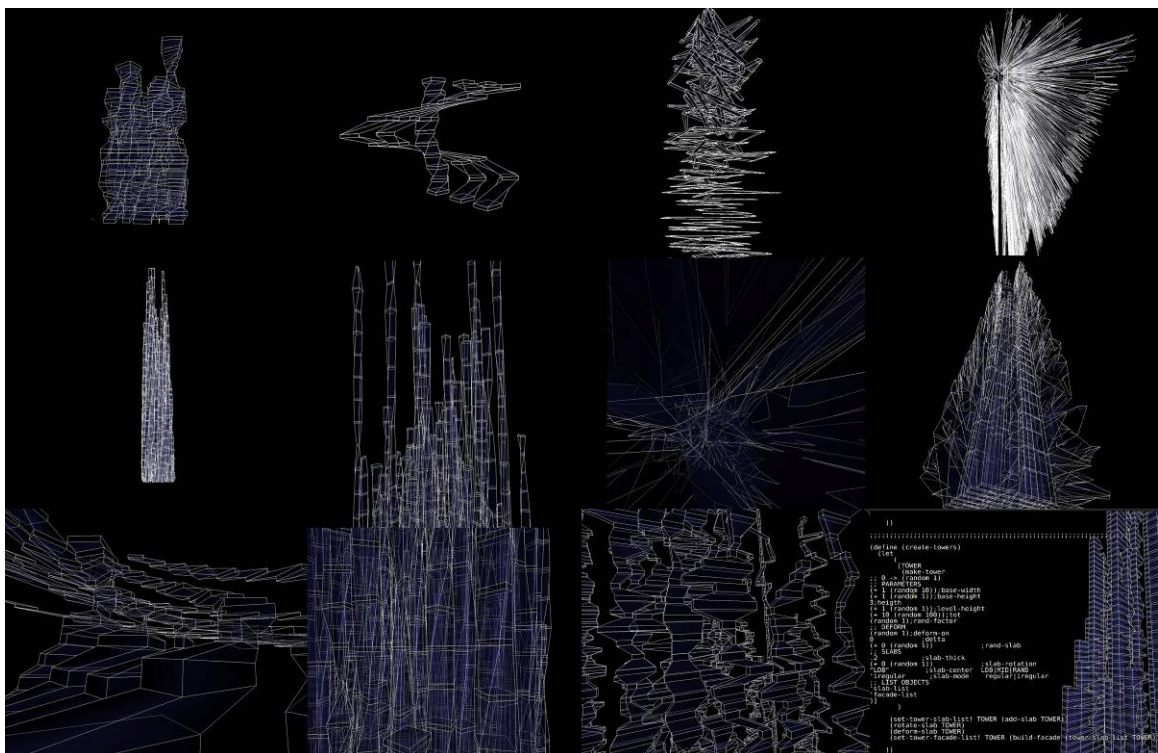


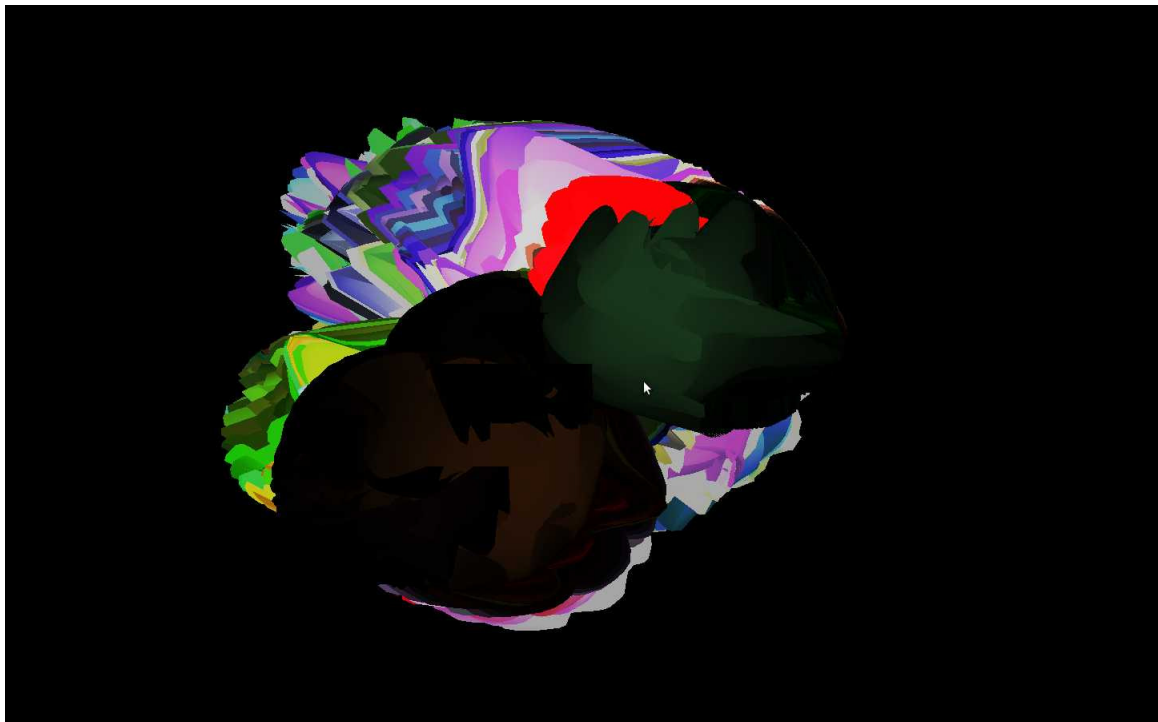
Figure 163: Série d'instantanés issus de l'expérimentation pour le concours *evolo tower* 2009

### **L'aléatoire et l'inattendu.**

L'aléatoire intégré dans le code est artificiel : il est provoqué par l'artiste et calculé par la machine. Cependant le décalage entre ce que l'artiste imagine et ce qui est exécuté crée une autre forme d'inattendu. Celle-ci n'étant non pas issue de calculs mais d'accidents. « *Rien ne se passe jamais comme on l'imagine* » [Yanatchkov, 2010]. Contrairement à ce que l'on pense de l'implacable logique d'un programme, certaines choses échappent au contrôle du domaine des solutions. Bugs ou accidents, ils proviendraient d'erreurs dans le programme, d'inattendus dans les rencontres de variables, etc. Comme si le code provoqué par M. Yanatchkov résistait et réagissait (Comme le ferait un « individu technique » ? [Simondon, 1958]).

L'inattendu peut aussi se loger dans l'observation d'instantanés extraits du mouvement des objets créés. Lorsque l'artiste provoque des mouvements ou des transformations continues d'objets, il peut décider à tout moment « d'arrêter le temps » et de figer l'image (cf. Figure 164). Se découvre alors quelque chose de caché, d'inattendu, dans lequel l'artiste va reconnaître des couleurs, des formes, des figures, qu'il appelle ses « peintures ».

Pour le concours *evolo tower 2009*, M. Yanatchkov a dû répondre à la contrainte de présenter une image fixe. Au travers de l'exploration de variations de tours (cf. Figure 163), M. Yanatchkov distingue des singularités et identifie des formes pour les dégager du flux.



**Figure 164 : les « peintures »**

### **Dépasser les contraintes, brouiller les frontières**

L'œuvre ne peut pas être circonscrite et exposée traditionnellement. Elle doit être appropriée par l'exploration des processus. Ces nouvelles pratiques s'accompagnent d'une désacralisation de la figure

de l' « artiste » : les codes de r,v,b,a sont disponibles sur Internet, et (fluxus) est « libre », c'est-à-dire mis en ligne, appropriable et modifiable dans sa structure même : sa source. Le « libre » (ou « open source ») est un rapport alternatif aux logiciels. Un logiciel libre est le fruit du travail d'un individu ou d'une communauté partagé librement pour permettre à quiconque de se l'approprier, de l'améliorer et de le modifier. Grâce en particulier aux logiciels libres et aux communautés qui les portent, le « design dans le code »<sup>113</sup> provoque des processus toujours plus fluides, entremêlés et ouverts. Milovann Yanatchkov revendique cette remise en avant du « lien » sous toutes ses formes : entre les membres d'une communauté développant un programme libre, entre le développeur d'un logiciel et l'architecte, entre l'artiste et potentiellement quiconque sur la Toile qui serait intéressé pour poursuivre son travail ou le réinterroger.

Cette pratique artistique a fait voler les contraintes usuelles liées à la temporalité de l'œuvre. Lorsque Dave Griffiths ou Milovann Yanatchkov font du livecoding, le temps de la réception de l'œuvre est différent de celui de la performance puisque l'œuvre c'est aussi (fluxus) lui-même. Finalement le « design dans le code » transforme à la fois le temps de la conception et celui de la réception de l'œuvre. La réception de l'œuvre se fait dans la durée, dans l'expérimentation du processus (ici au sens informatique, c'est-à-dire une expérimentation du code). Le temps de la conception ne peut être circonscrit et se situe dans l'inachevé.

On assiste ici à une expérience marginale de l'architecture. À travers cette pratique, c'est une réelle transformation du mode de conception qui est pointée. On prend conscience du bouleversement de ce qu'implique le « design dans le code » tel que John Maeda l'appelle de ses vœux<sup>114</sup>.

On l'a vu, les pratiques de r,v,b,a comme celles de Dave Griffiths brouillent nos catégorisations habituelles. S'agit-il d'un travail d'architecte ? d'artiste ? de hacker ? On ne sait plus où sont les limites: le processus devient le cœur de l'architecture, la forme un événement, un logiciel une œuvre d'art.

## **2.14 Marquise Gare d'austerlitz, AJN (Paris, 2010-2011)**

### Description du projet et du modèle paramétrique développé

Le projet d'intervention sur la Gare d'Austerlitz est une collaboration entre AREP et AJN. La marquise à laquelle nous nous intéressons couvre l'entrée de la gare. Selon T. Kubota (Kubota 2012), cette marquise a deux visées : celle de protéger les voyageurs de la pluie (échelle fonctionnelle) et celle de faire « signal » (échelle optique). Cette marquise est conçue en référence à la « fumée du train » (échelle de modèle). L'utilisation du verre est venue appuyer l'idée de transparence, s'est alors posé la question des dimensions du calepinage du verre. Un calepinage a été proposé pour à la fois assurer l'étanchéité de la marquise (échelle technique et fonctionnelle) et répondre aux carreaux des vitraux existants (échelle de voisinage et de modèle). Il s'est avéré difficile pour les architectes de

---

<sup>113</sup> « *Design dans le code* » est le nom d'une exposition réalisée par John Maeda à la Fondation Cartier en 2006.

<sup>114</sup> [plw.media.mit.edu/people/maeda/]

faire se superposer à la forme déjà conçue de la marquise un calepinage composé de carreaux de dimensions identiques. L'utilisation de Grasshopper a permis à la cellule 3D d'optimiser des tailles de carreaux pour avoir un seuil de maximum 3cm d'écart entre le calepinage des carreaux et la forme de la marquise. La modélisation paramétrique a donc ici surtout servi à évaluer les performances et les caractéristiques géométriques de l'ouvrage.

Finalement ce projet de marquise en calepinage en verre n'a pas été conservé.

### Opérations de modélisation paramétrique et de conception architecturale

| Opérations de Conception Architecturale   | Relation  | Opérations de modélisation paramétrique   |
|---|---|---|
| Dimensionner la géométrie pour « Faire signal »<br>échelle optique  | — est suivi de →<br><br>(la modélisation est ici une représentation d'objet déjà conçu) | Dessin d'une surface reprise depuis Rhino   |
| Dimensionner la géométrie de la marquise comme « une fumée de train »<br>échelle sémantique                                 |   |   |
| Dimensionner la marquise pour qu'elle puisse protéger les voyageurs de la pluie<br>échelle fonctionnelle, échelle technique | — est suivi de →  | L'exigence d'étanchéité implique la détermination d'un seuil de max 3 cm entre les carreaux |

Peut-on observer des « niveaux de conception » ?

Données non disponibles.

Peut-on observer des opérations logiques liées à la modélisation paramétrique ?

Données non disponibles.

Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?

L'utilisation de Grasshopper ici a été déterminée par des choix de conception fait en amont, des opérations de collaboration sont probablement intervenues pour assister la modélisation paramétrique.

## 2.15 City Hall, Foster and Partners (Londres, 1998-2002)

Description du projet et du modèle paramétrique développé

Pour ce projet, le SMG (cf. infra XXX) et l'équipe de conception ont collaboré dès les premières phases de la conception pour formaliser, ensemble, la volumétrie de l'édifice. Le modèle demandé au SMG devait permettre générer rapidement des instances à évaluer en termes d'ensoleillement. L'édifice avait été pensé «by the river » (Kolarevic 2005, p.83) par l'équipe de conception et interprété par le SMG en terme de «galet paramétrique » (« parametric pebble », (Kolarevic 2005)). Les



intentions déterminantes du projet ont conduit le SMG à produire différents modèles paramétriques suivant l'avancement du processus de conception. Ces intentions ont été de :

1- faire référence à un « galet » : pour cela le volume a été construit à partir de surfaces de type nurbs, qui contraignent les possibles déformations du modèle, définies par un nombre réduit de points de contrôles, afin de garder une courbure de surface peu marquée ;

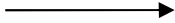
2- contrôler l'ensoleillement du volume en créant le plus d'ombre propre possible : ceci a conduit à proposer une composition le long d'un axe ;

3- Composer une symétrie est-ouest du volume, correspondant à une symétrie faisant face à la Tamise. L'asymétrie nord-sud peut, quant à elle, être expliquée par la volonté de contrôler l'ensoleillement, voir 2-.

4- Créer des proportions géométriques contraintes en relation à la référence à un « parametric pebble ».

Plusieurs modèles paramétriques ont été conçus à différents stades du projet, lors desquels le SMG a participé à la pensée architecturale en : 1) proposant un découpage du volume par des plans de niveaux, 2) proposant une géométrie acoustiquement optimisée de l'atrium central et 3) calculant les surfaces développables de l'enveloppe pour les rendre constructibles.

L'analyse des opérations cognitives en jeu dans ce cas de conception ne permet pas de disjoindre clairement les mécanismes cognitifs de la conception architecturale et de la conception du modèle paramétrique. Ici, intentions architecturales et propositions du SMG composent une complexité opératoire qui fait la conception architecturale même. Bien que le SMG se disjoigne des équipes de conception, ses interventions ont donc ici participé pleinement de la conception architecturale.

| Opérations de Conception Architecturale  | Relation  | Opérations de modélisation paramétrique   |
|--|---|---|
| Penser le projet « by the river »<br>Echelle sémantique ?  | <br>relais généré par<br>l'opération<br>d'interprétation (?) | Proposition de découpage du modèle en nurbs en référence au « parametric pebble » |
| Contrôler l'ensoleillement pour obtenir un maximum d'ombre propre<br>Echelle technique et géographique |   | Dimensionnement du volume autour d'un axe orienté en fonction de l'ensoleillement |
| Donner au bâtiment une façade principale<br>Echelle de visibilité                                      |   | Mise en relation de deux cotés du volume <i>via</i> une symétrie                  |

Peut-on observer des « niveaux de conception » ?

Il y a ici deux niveaux de conception exploités par des acteurs différents : la modélisation paramétrique est effectuée par le SMG, l'instanciation est effectuée par l'équipe de conception.

Peut-on observer des opérations logiques liées à la modélisation paramétrique ?

Données non disponibles.

Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?

Etant donné le contexte alors novateur de la modélisation paramétrique, il est probable que le SMG ait été à la fois force d'interprétation et de proposition face à l'équipe de conception.

La collaboration est ici majeure à prendre en compte : non seulement elle a « déterminé » la modalisation paramétrique mais elle a également influencé le modèle paramétrique pour rendre l'interface la plus simple possible à utiliser pour l'instanciation.

# Analyse architecturologique de cas documentés par les expérimentations

## 2.16 Projet C1, Toshi et Pierre (Expe1)

Description du projet et du modèle paramétrique développé

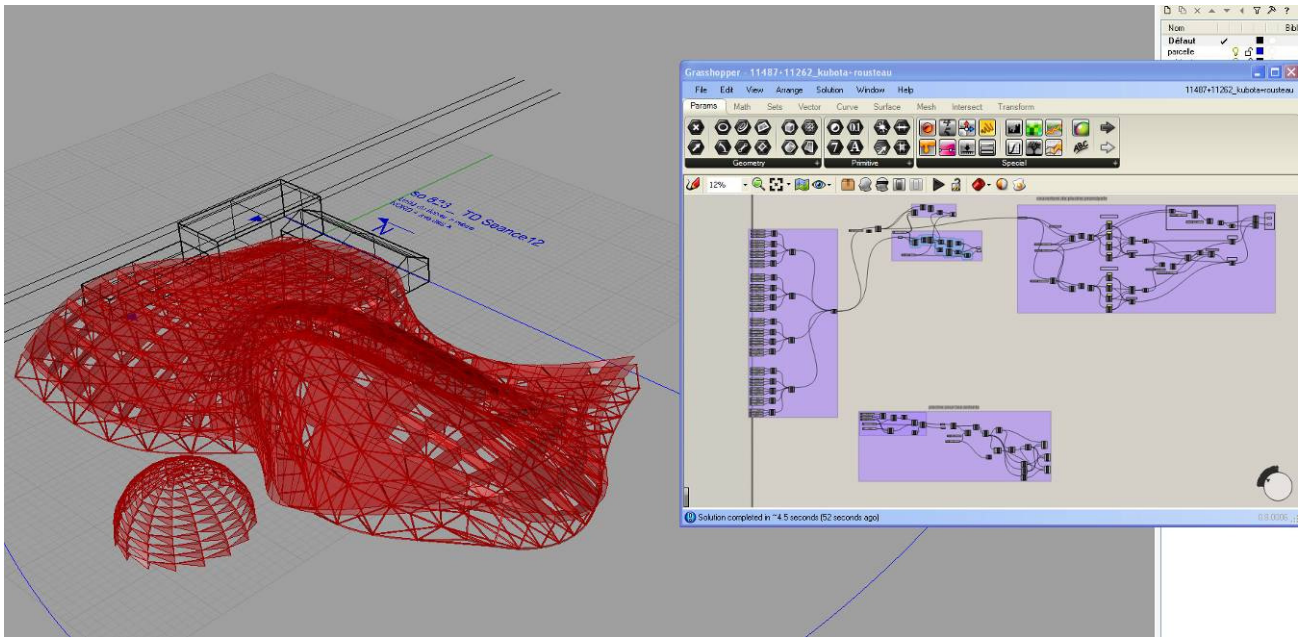


Figure 165 : Projet C1 (EXPS0821-2011), vue générale, sources : A. de Boissieu d'après les modèles de T. Kubota et P. Rousteau

Dans un premier temps les étudiants ont modélisé sur Rhino les surfaces souhaitées pour « *trouver un accord en binôme* » (cf. note d'intention). Une fois un accord trouvé, cette surface a été traduite sur Grasshopper via une opération de découpage du modèle de la surface en une surface par section (loft) s'appuyant sur des polygones définies par des points (cf. Figure 167). Les variables les plus importantes (champ de valorisation) sont ici les coordonnées numériques des points, ce qui est critique quant aux transformations possibles de l'instance.

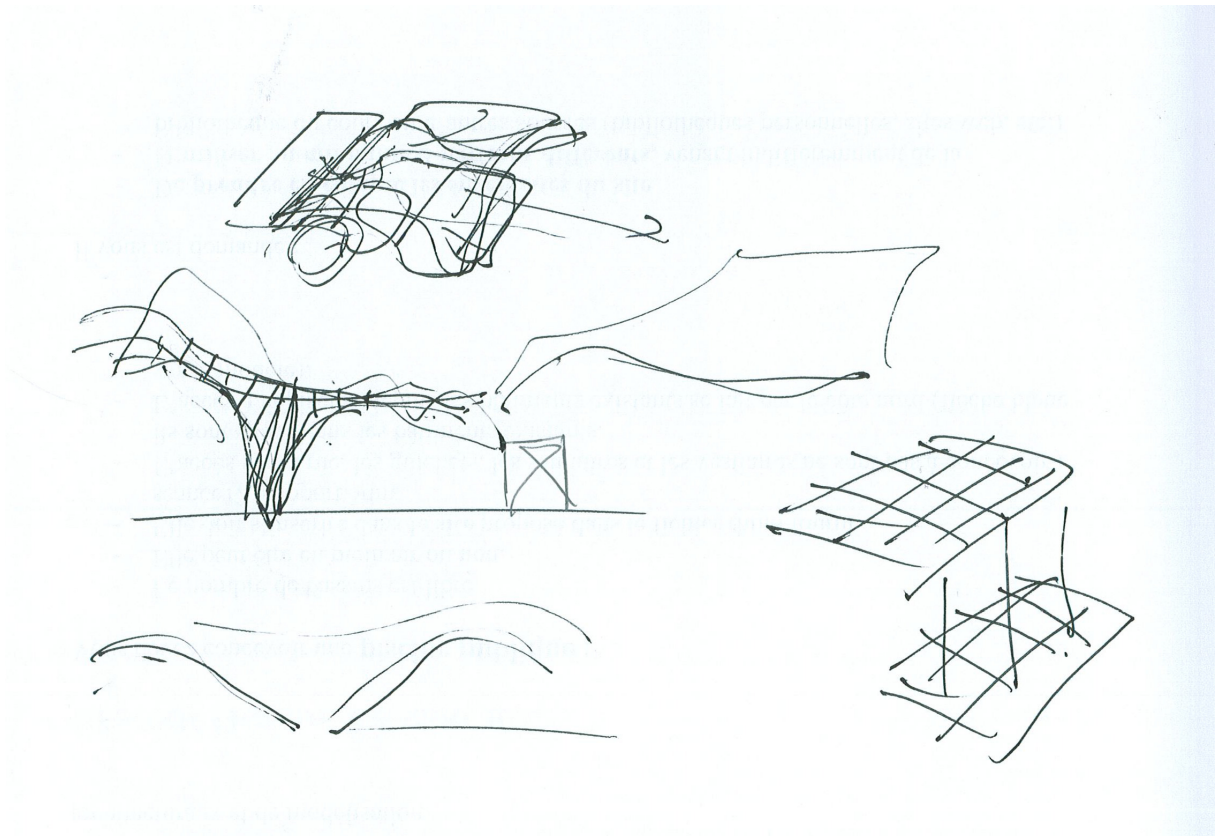


Figure 166: esquisses du projet du groupe C1 (EXPso821-2011)

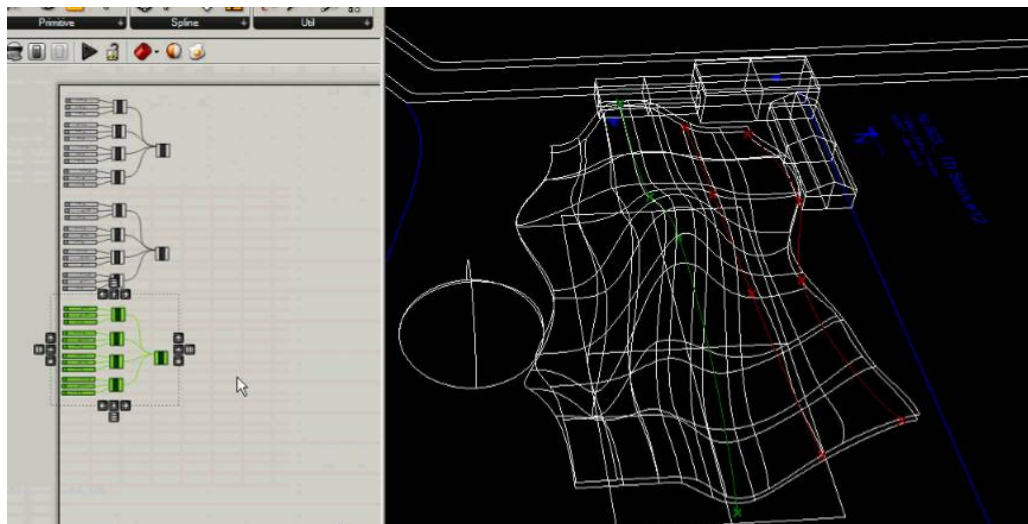
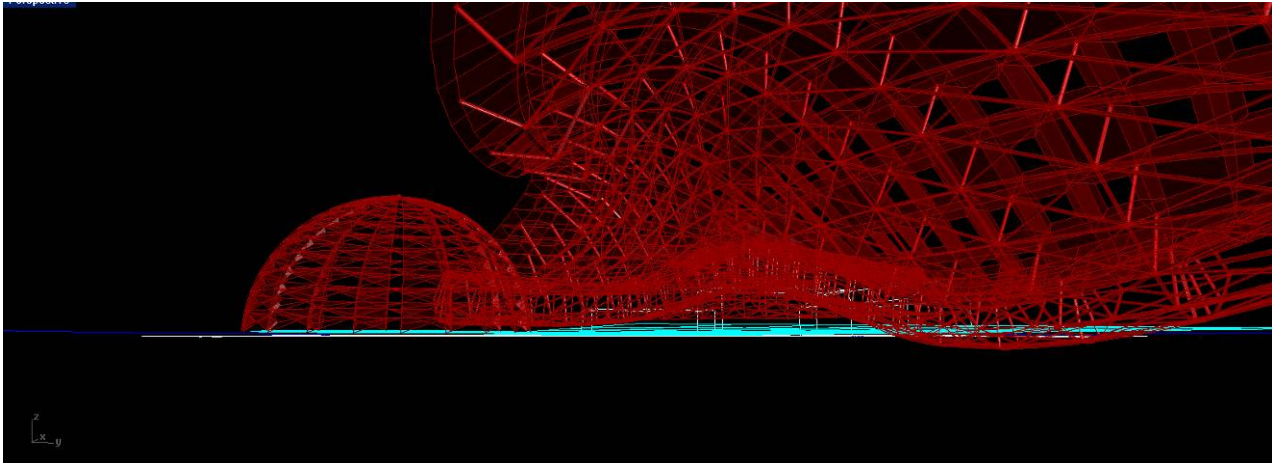


Figure 167 : Premières étapes de modélisation du projet C1 (EXPso821-2011), capture d'écran issue du film de l'expérimentation, sources : T. Kubota et P. Rousteau

Cette surface est soumise à deux translations en  $z$  générant deux surfaces. Ces surfaces sont utilisées comme entrées à deux samples : l'un générant la structure (échelle technique), l'autre générant l'enveloppe « signal » (échelle optique).

Le premier sample est « créer\_motif-maillage ». Dans ce sample la variable « rayon » des barres est modifiée manuellement (aucun rapport n'est établi dans le modèle entre la longueur des barres par exemple et le rayon). Les deux nappes sont liées par des barres créées entre des points générés de façon homologue (échelle technique). Sur la surface est également « appliqué »<sup>115</sup> le sample « evaluer-planeite-surface », qui ne sera pas utilisé par la suite, et le sample « Panels-dispatch ».

Le choix de deux « motifs » a amené les étudiants à faire deux « pavillons ». Le deuxième pavillon est une piscine pour enfant, dimensionné comme un dôme, et couvert d'un motif « en pomme de pin » (cf. enregistrement audio) issue de la bibliothèque de modèles de Toshi.



**Figure 168 : Perspective depuis la piscine vers la « vue dégagée » et le bassin pour enfants, sources : T. Kubota et P. Rousteau**

| Opérations de Conception Architecturale  | Relation  | Opérations de modélisation paramétrique  |
|--|---|--|
| Dimensionnement de l'enveloppe générale du bâtiment pour créer une liaison avec l'existant (échelle de voisinage) et pour ouvrir l'espace intérieur sur la « vue dégagée » (échelle de visibilité) | — est suivi de →<br>(la modélisation est ici une représentation d'objet déjà conçu) | Dessin d'une surface reprise depuis Rhino, surface dessinée à partir de points, liés en polylignes, liées en un « loft » |
| Dimensionnement du projet en deux pavillons.   | — est suivi de →<br>(le projet a été influencé par une opération de modélisation)   | Choix d'utilisation de deux samples « motifs » (échelle de modèle ?).  |
| Création de deux « nappe » pour distinguer l'enveloppe de la structure   | — est suivi de →<br>«P : « mais tu crois qu'on peut le faire ? » T : « oui oui »»   | Découpage du modèle en une surface principale copiée avec un « offset »  |
| Dimensionnement d'une structure treillis   | ↔   | « Application » du sample « créer motif  |

<sup>115</sup> Le terme « application » est utilisé par les étudiants dans leur fiche d'intention.

|   |                       |   |
|---|-----------------------|---|
|   | traduction            | maillage » sur la nappe inférieure (éch de modèle ?), instanciation de la variable « rayons » des barres du treillis (ech technique)    |
|   | ←<br>est suivi de     | Difficulté avec l'utilisation des points de la surface (enregistrement audio, minute 59')   |
| Dimensionnement d'une enveloppe « signal » (échelle optique, échelle de modèle) | ↔<br>surdétermination | « Application » du sample « Panel Dispatch » sur la nappe supérieure. Sample choisi pour : les jeux d'ombre, l'esthétique (ech optique) |
| Liaison des deux nappes (ech technique)   | ←<br>est suivi de     | Raccord des deux surfaces par des barres (Toshi : 33' « c'est un classique de Grasshopper »)  |
| Dimensionnement d'une enveloppe pour le pavillon du bassin pour enfants (ech ?) | ↔                     | « Application » d'un sample venant des documents personnels de Toshi.   |

Peut-on observer des « niveaux de conception » ?

Il existe un jeu entre l'instanciation et la modélisation paramétrique.

Peut-on observer des opérations de modélisation paramétrique spécifiques ?

|                               |   |
|-------------------------------|---|
| Opérations de découpage       | Découpage de la surface en polygones dirigées par des points (opération de hiérarchisation ratée) |
| Opérations de dimensionnement | Dimensionnement des variables numériques et des surfaces d'entrée des samples                     |

Peut-on observer des opérations pragmatiques liées à la modélisation paramétrique ?

Les jeux entre instanciation et modélisation paramétrique sont ici flagrants. En particulier le passage entre l'instance « surface » créée sur Rhino et le modèle Grasshopper créé ensuite exemplifie une opération d'inférence (passage du particulier au système général). Cependant le but de la modélisation est ici surtout d'obtenir une surface identique à celle choisie (sur papier puis validée sur Rhino), la généralisation en modèle paramétrique s'est avérée pauvre. En effet la surface est transformée en modèle paramétrique *via* une liste de points définis par des coordonnées difficilement manipulables (48 sliders créés pour contrôler une surface).

Peut-on observer des opérations de collaboration liées à la modélisation paramétrique ?

Il y a eu peu de collaboration et de négociation entre les étudiants. La conception s'est jouée dans des échanges de propositions / validations entre les concepteurs. L'étudiant qui possédait la main



(Toshi) ayant une bonne maîtrise de Grasshopper s'est retrouvé majoritairement maître du projet (cela s'observe *via* l'absence d'interactions perceptible dans l'enregistrement audio).

## 2.17 Projet D1, Victoire et Aymeric (Expe1)

Description du projet et du modèle paramétrique développé

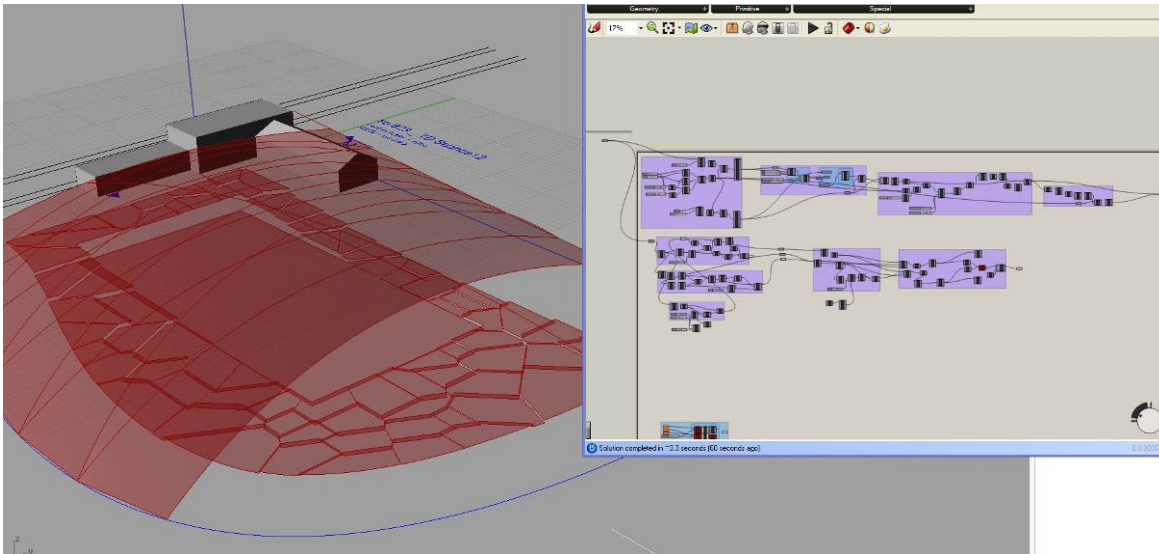


Figure 169 : Projet D1 (EXPso821-2011), vue générale, sources : A. de Boissieu d'après les modèles de V. Saby et A. Perret du Cray

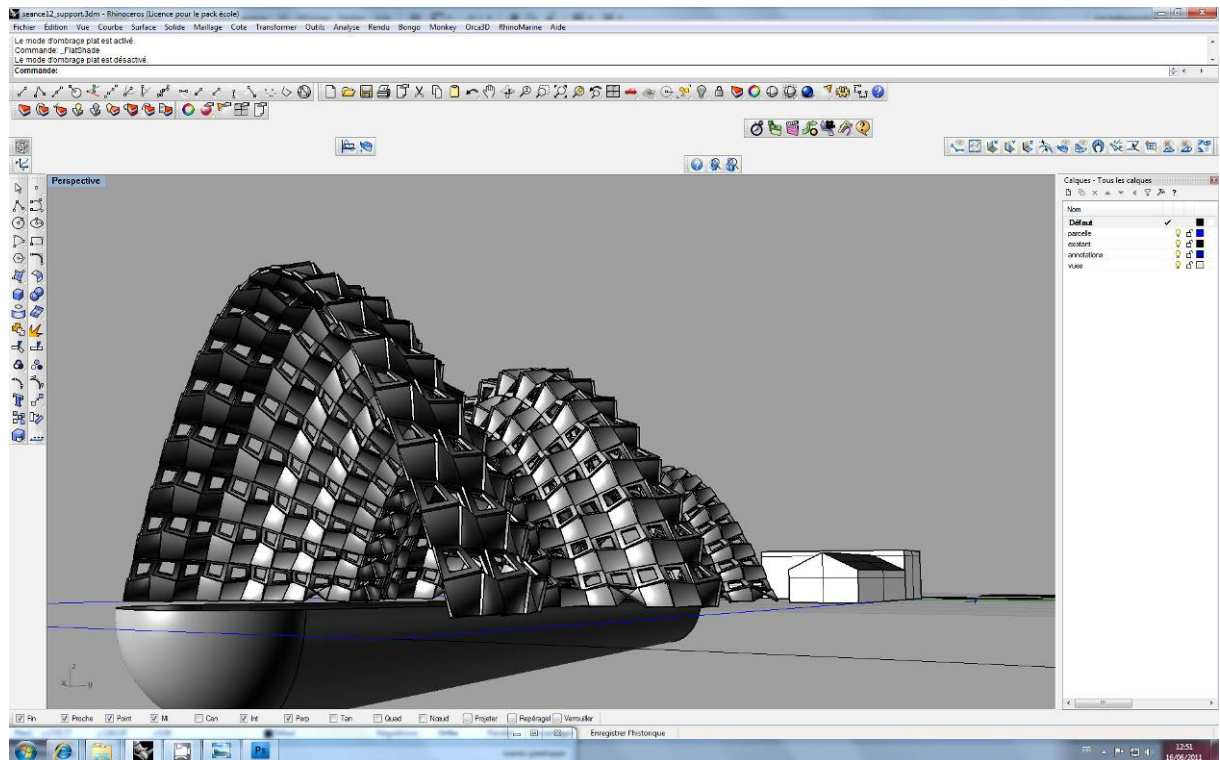
Le projet D1 s'organise autour d'une piscine principale pour nager (échelle fonctionnelle) à laquelle s'associent des bassins peu profonds « pour jouer avec les enfants »<sup>116</sup> (échelle fonctionnelle, échelle humaine). Les petits bassins se déversent dans le grand. L'emprise totale des bassins est comprise dans un « offset » de la parcelle (échelle parcellaire)

## 2.18 Projet A2, Rebecca et Sebastiaan (Expe1)

Description du projet et du modèle paramétrique développé

Le processus de conception du groupe A2 semble pouvoir être lu *a posteriori* comme s'organisant en quatre phases : - la conception de la forme de l'enveloppe; - la conception du « treillis »; - la conception du bassin de la piscine ; et - la conception des ouvertures, implantées sur le treillis.

<sup>116</sup> Expression utilisée par les étudiants dans la note d'intention.



### Projet A2 (EXPso821-2011), extrait du film de captures d'écran

Lors de la conception du treillis, modélisation et conception ont été menées de front et, les étudiants ont eu recours à un pattern. Cette phase a été identifiée et délimitée : elle dure treize minutes et fait l'objet d'une analyse architecturologique détaillée (cf. tableau 4b inséré en annexe).

Dans le cas du travail du groupe A2, la présence d'une bibliothèque de patterns ainsi que de l'obligation d'utiliser au minimum trois patterns a induit une échelle de modèle. Le « treillis » a tout d'abord été dimensionné en fonction des patterns disponibles [c1, c2].

[c1]

Rebecca « on se fait un treillis un peu comme ce qu'on avait vu là sur le site hum... [silence] [clics]  
Ouais faut regarder les patterns dans... ici là »

[c2]

Rebecca [bruits de papiers] « En fait on peut faire un mix entre ça et ça ? »

### Retranscription, Extrait

#### Dimensionnement de samples

L'échelle optique dans le cas du groupe A2 a induit des opérations de modélisation paramétrique. Ainsi la volonté de « ne pas casser la forme » en utilisant un motif régulier [c5] a amené à sélectionner un pattern divisant régulièrement la surface de référence en sous surfaces [mp5].

[c5]

S non non non on va casser la forme si on fait ça.  
R bah ça dépend si c'est régulier on va rien casser du tout.



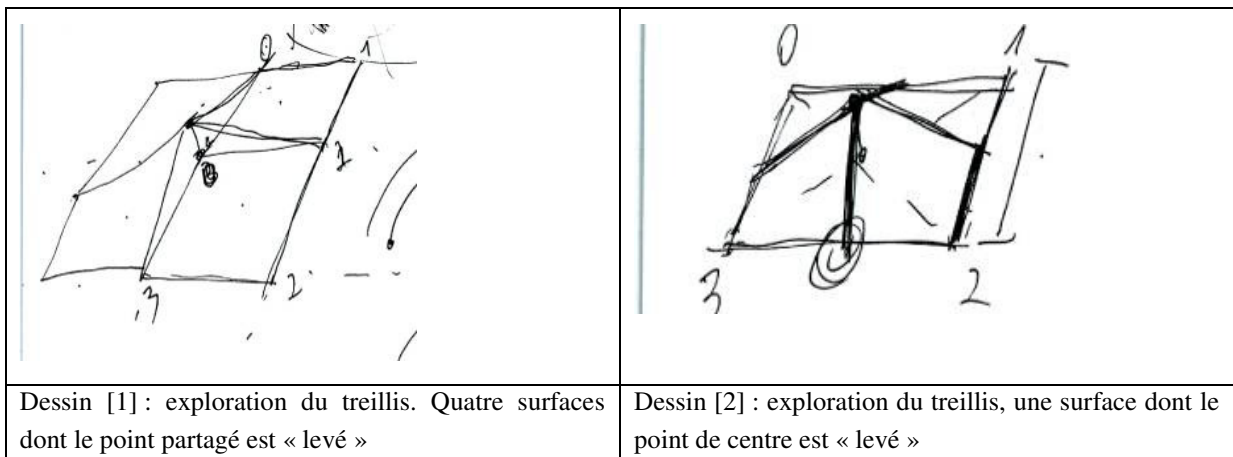
[mp5]  
 R Ça va être terrible !  
 On commence par diviser la surface en parties régulière...  
 S euh.. On fait comment... Je crois que se serait le plus facile avec euh...  
 R Comme ça [bruit de tapotement sur la table (probable désignation d'un pattern)] En des sous-domaines avec...  
 S Oui, il y a deux moyens j'ai vu, attend... ici [clic\*\*]  
 R C'est le pattern « créer un réseau de points »  
 S hum... j'ai vu, je crois que j'ai vu... [clic] Comme ça ! « sous-surfaces » et on fait un « explode » après. Comme ça tu as à chaque fois un carré et tu peux après ...  
 R Oui, ok

### Retranscription, Extrait

Le pattern choisi est adapté en fonction de l'échelle de modèle [c4], en effet les étudiants ne veulent pas un treillis « à plat comme dans le cours », mais souhaite lui donner une géométrie approchant celle de leur référence de Félix Candela. Cela les amène à questionner leurs intentions [mp7], à optimiser le modèle pour le rendre performant et ne pas multiplier inutilement les données à traiter [mp6]. Dans cet extrait l'optimisation du modèle et en général un certain pragmatisme intervient souvent dans la conception.

[mp6]  
 S hum... attend. Faut bien le penser. Comme ça tu as quatre carrés [dessin 1] Si c'est 1, 0, 2, 3 ; probablement pour le prochain... 0, 2 ... [rire] Ok, se sera un peu trop difficile. Parce que pour chaque petite surface se sera un autre point.  
 R Oui mais est-ce qu'on ne peut pas faire un ensemble de quatre surfaces à chaque fois ? Est-ce qu'on ne peut pas réunir...  
 S Ah oui ! Et on prend le milieu et hop !  
 R ce sera plus facile...  
 S Evidement  
 R Sinon là on y est demain encore, c'est pas possible ! [rire]

### Retranscription, extrait

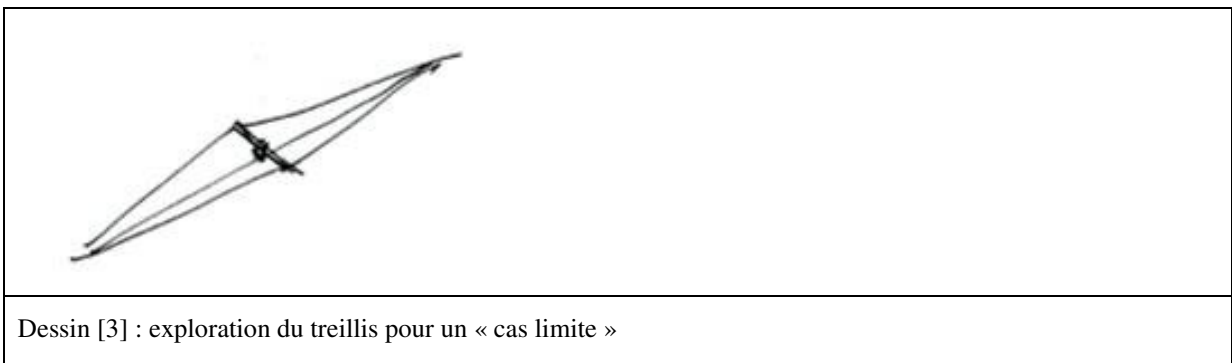


Esquisses, Extrait du tableau 4c inséré en annexe

L'échelle géométrique semble importante dans le cas étudié, opération de conception et opération de modélisation s'imbrique pour penser la géométrie du treillis, en particulier pour penser le point de centre [mp10].

### Sample et opération logique et pragmatique

On observe dans cet extrait un phénomène intéressant car symptomatique du mode de pensée général/particulier que nécessite la modélisation paramétrique. Une fois le treillis conçu [mp6] un étudiant émet un doute et éprouve le besoin de tester le système sur une surface « limite » pour en éprouver la capacité à être vraiment générique [mp9] [dessin 3].



### Esquisses

## 2.19 Projet A2, Rebecca et Sebastiaan (Expe 1), CODAGE ET RETRANSCRIPTION

### 2.19.1 Analyse globale de l'expérimentation de 9h47 à 11h45

| Time line    | Paroles (extraits)  | Manipulation GH | Ecriture, dessin                 | Commentaires | Première analyse  |                                |
|--------------|---|-----------------|----------------------------------|--------------|---|--------------------------------|
| 1'00<br>9h47 | [discussion sur le toit] ...<br>S« pas trop joli » ...<br>S« c'est bien parce que si on fait un arc c'est structurel » ( ...)   |                 | Dessin toiture<br>« coupe » ( ?) |              | Dessin avant modélisation. Premiers dessins avant discussion.   | (A) Discussion sur la forme de |
| 2'28         | R« c'est bien peut être qu'on peut commencer » S« bah on peut commencer par définir l'axe » R« vers la vue dégagée » S« axe sur rhino et les points on peut les définir |                 | Dessin « persp »                 |              | Dessin de l'idée de la forme de la piscine avec ref stucturelle et esthe plutôt que rapport au site. Rapport au site vient au moment de |                                |

|               |  |  |        |   |  |  |
|---------------|--|--|--------|---|--|--|
|               | comme ça »   |  |        |   | modéliser « on peut commencer » [la modélisation]. |  |
| 3'            | Spline sur rhino versus point sur GH => S choix points sur GH car meilleur contrôle S « courbe interpolé », S« avec paramètres ici », R« avec 1, 2, 3... 8 points...ça dépend de la taille de la piscine qu'on veut »  |  |        | Compte le nbr de point à faire sur la persp                                   |  |  |
| 3'50<br>9h50  | S« on va regarder » « hop, hop »<br>R « on se fait une ligne comme ça là » S « comme ça là ? » R « un peu plus vers le nord, ouais parfait »   |  |        | Clic, ouverture du fichier, dessin d'une ligne sur Rhino, Silence, juste clic | Instanciation de l'idée sur le site                |  |
| 4'40          | S « ok on commence avec une courbe » points « avec paramètre ou 1/6° ? » R « oui ce sera plus simple » divise [nom de composants] silence « voilà » [silence + commentaire de GH]  | Ouverture GH<br>Modélisation des points comme division de la courbe en 6 |        |   |  |  |
| 7'18          | S« ok maintenant il faut trouver la normale » R« la distance » S explore, repère un composant, l'explique à R. continue a explorer, parle pour lui, clic, silences   | Modélisation des normales aux points précédents                          |        |   |  |  |
| 9'16          | R et S « parfait »<br>R «maintenant faut donner des distances, ce qui serait chouette c'est que ça monte de plus en plus » S « oui on peut mettre des paramètres » discussion « ah oui » intentions R « moi j'ai pensé ... « il faut prendre ce point là, ce point là, et prendre une hauteur » « il faut... » « non c'est les mêmes... »<br>R« c'est pas terrible ça » S« on va voir si on y arrive » |  | dessin | Négociation, puis traduction en GH  | Conception de l'arc/du rail                        |  |
| 11'50<br>9h58 | Silence<br>R « ok nickel »<br>S « ok on va organiser un peu.. ok » « maintenant on a un vecteur »<br>R« on donne quoi comme distance ? on va voir ce que ça donne » S« 27-25 ? » R «on peut commencer avec 20 »  | modélisation   |        |   | Modélisation des points du rail                    |  |
| 15'02         | R « ok, maintenant il faut   | Exploration  |        | R donne les   | « Instanciation » des                              |  |

(B) conception enveloppe

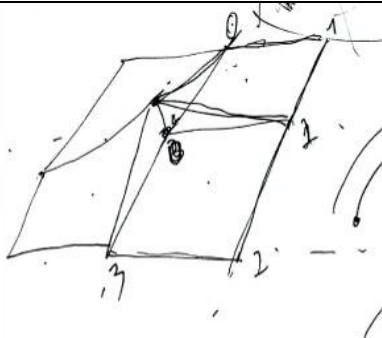


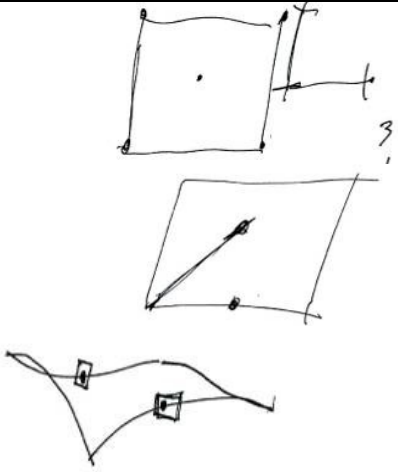
|                |  |  |        |   |   |  |
|----------------|--|--|--------|---|---|--|
| 10h01          | changer les distances »<br>« le premier on le met à 20, le deuxième un peu plus petit »<br>Silence<br>R « peut être qu'il faut réduire les distances ? on verra après »<br>silence   | des valeurs des sliders  |        | valeurs, S modélise   | rails de toitures   |  |
| 18'<br>10h04   | « miroir »..<br>Silence<br>R « oui ça marche bien » « ah non ça marche pas »   | Miroir des points, courbe interpolée   |        | S manipule, R regarde et valide et commente                                   |   |  |
| 20'41          | R « est-ce que ça t'embêterait si on déplace la courbe » « l'axe qui conditionne tout pour mettre les courbes dans l'axe du bâtiment » S « ah oui »<br>S « d'abord je modélise la surface »<br>R « la c'est bien c'est parfait, ça me plaît plus que le demi cercle  | Modélisation de l' « arc » comme une courbe interpolée<br>10h12<br>Modélisation de la surface par sweep2 |        | Exploration des composants arcs   | Modélisation de l'arc<br>Résistance de l'arc, ça ne marche pas,<br>exploration des autres solutions possibles |  |
| 26'10          | R et S « Cool » R « Y a que sur un logiciel qu'on peut faire ça, pas en vrai »<br>S « et maintenant on fait quoi ? »   | Organisation du modèle (groupe nommé)  |        |   |   |  |
| 26'47<br>10h13 | R « on fait un treillis euh... »<br>« un peu comme ce qu'on avait vu sur le site » « faut regarder les patterns »<br>R « on fait un maillage »<br>S « mais après le maillage on fait quoi ? parce qu'il faut utiliser trois.. » S « on peut faire un truc un peu bizarre avec les treillis » R « entre ça et ça ? »<br>S « non on va casser la forme » |  |        |   |   | (C) Exploration, conception et modélisation d'un composant répété sur la surface |
| 29'46          | S « ça me fait penser à quelque chose »<br>...<br>« treillis régulier » « c'est pas avec un attracteur alors » « non c'est autre chose » « il faut diviser la surface en forme régulière » « trouver la normale au centre de la surface »  |  | dessin | Exploration ref felix candela<br>Exploration dessin puis déduction méthode GH |   |  |
| 33'            | R « c'est le pattern créer un réseau de points » S « attend il faut bien réfléchir » R « est-ce qu'on ne fait pas un ensemble de 4 surfaces à chaque fois ? »  |  | dessin | Exploration support de cours pour réseau de points                            | Op de mod para. Qu efficacité   |  |
| 36'            | « subdivise »  | Divide Explode   |        | Division de la surface  |   |  |
|                |  |  |        |   |   |  |


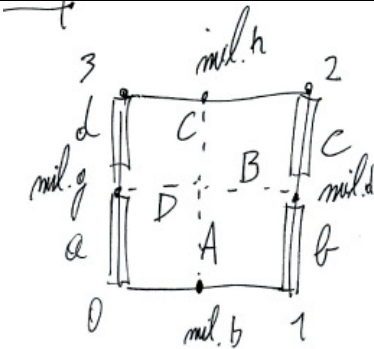
|                     |   |   |                               |   |  |     |
|---------------------|---|---|-------------------------------|---|--|-----|
|                     |   | item  |                               | Explode,<br>extraction des<br>points                  |  |     |
| 39'                 | S « Attend parce que si j'ai une surface comme ça est-ce que c'est toujours le milieu »                                       |   | Dessin schema<br>sous surface |   | Dessin / choix   |     |
| 40'                 | S « Est-ce bien nécessaire que ça se trouve au centre des surface » R« y a un moment dans un exercice où on a réussi »        | Exploration<br>pour trouver<br>un composant<br>qui extrait une<br>normale au<br>centre des<br>sous surfaces |                               | Quasiment<br>10min de<br>recherche/bloca<br>ge sur GH | modélisation   |     |
| 49'                 | S « Je crois que j'ai peut être trouvé »<br><br>53' R « oui ça le fait c'est nickel, fait voir de l'autre coté si ça marche » | 2-<br>modélisation<br>des points  | 1- dessin                     |   | S à force d'explorer (il a la main sur GH) finit par trouver un moyen (géométrique/informatique ?) et explique la possibilité à R à l'aide d'un schéma |     |
| 53'                 |   | Après<br>modélisation<br>des points,<br>images des<br>points pour<br>treillis                               |                               |   |  |     |
| 11h15               |   |   |                               | (fin de la surface<br>à 11h15)                        |  |     |
| 11h15<br>-<br>11h38 |   |   |                               | Modélisation du<br>bassin de la<br>piscine            |  | (D) |
| 11h42               |   |   |                               | Début de<br>l'exploration du<br>morphing              |  | (E) |

### Analyse globale, succincte et linéaire de la vidéo

| tps | Ouverture d'un site de patterns        | Recherche d'autres références internet              | commentaires |
|-----|--|---|--------------|
|     | 10h13 patterns so823                   |   |              |
|     |  | 10h16 google « felix candela »                      |              |
|     | 10h19 so823 cours séance 2             |   |              |
|     | 11h42 so 823 cours séance 3 (morphing) |   |              |
|     |  | 11h44 googl « felix candel » puis page web reinhaus |              |
|     | 12h17 cours séance 3 (morphing)        |   |              |

## 2.19.2 Analyse des dessins rendus par les étudiants, relatifs à la séquence analysée dans le tableau 2.19.3

| code       | extrait   | Commentaires  |
|------------|---|---|
| Dessin [1] |    | Exploration du treillis.<br><br>1- Division de la surface en « carrés » (ici 4 carrés visibles)<br>2- Les points se lèvent<br>3- Question ordre des points... → problème car le point levé = 4 points !<br>=> décision de réunir les 4 surfaces (différence visible dans le dessin [2])             |
| Dessin [2] |   | Exploration du treillis.<br><br>34'00 S « Il faut d'abord penser, [bruit de papier et de dessin] tu prends une surface, tu prends le milieu, le point, on le lève, et on fait ... point comme ça, point comme ça... »   |
| Dessin [3] |  | Exploration du treillis. Exploration de la géométrie<br>40'00<br>S Attend parce que j'ai dessiné un carré mais si on fait une surface comme ça, est-ce que c'est encore le milieu ? [dessin 3]<br>R Oui bien sur<br>Oui, un parallélogramme, normalement... il faut prendre là où les deux touches. |
| Dessin [4] |  | Exploration du treillis. Exploration de la géométrie  |

|               |   |  |
|---------------|---|--|
| Dessin<br>[5] |  | Exploration du treillis. Exploration de la géométrie |
| Dessin<br>[6] |  |  |

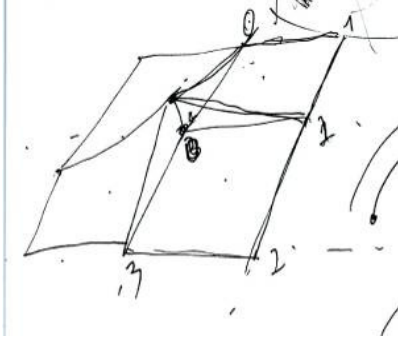
### 2.19.3 Retranscription et Analyse approfondie d'une séquence [10h13 à 10h36]


Le tableau suivant retranscrit l'analyse du passage de conception du « treillis » en référence à F. Candela, sur une surface déjà conçue (de 26'04 à 39'00, séquence de 13 min).

|                   |   | <i>Extrait de la retranscription</i>  |
|-------------------|---|---|
| Echelle de modèle | [c1, c2]<br>Conception du treillis au travers du choix de patterns dans la bibliothèque   | 10h13<br>[c1] Rebecca « on se fait un treillis un peu comme ce qu'on avait vu là sur le site hum... [silence] [clics] Ouais faut regarder les patterns dans... ici là »<br><br>[c2] Rebecca [bruits de papiers] « En fait on peut faire un mix entre ça et ça ? »<br><br>ATTENTION, la référence au programme montre une contrainte dans le choix des patterns<br>10h15 Sebastiaan « Il faut utiliser trois trucs »             |
|                   | [c4] une première exploration des références de treillis renvoie S à un projet de Félix Candela, ils en reprennent l'idée géométrique | 10h16 [c4]<br>S En fait ça me donne une idée. Une fois j'ai vu quelque chose. C'était une sorte de shell [clics]. Je vais chercher. Ça m'a fait un peu penser à ça... Ça faisait comme des ondes, comme une sorte de treillis et ipar (?). En chaque petit carré, il y avait un ipar.<br><br>R C'est exactement notre forme ! [forme : celle de l'enveloppe ndlr] [rire] Enfin en gros on la continue mais c'est la même chose. |

|  |  |   |
|--|--|---|
| Echelle géométrique                        | Faire un motif qui ne soit pas plat comme vu en cours, mais qui « se lève » comme dans la référence de Felix Candela | 10h15<br>Mais c'est... Qu'est ce que tu voulais dire en disant « en montant » ?<br>S On peut faire... Normalement c'est plat, dans le cours<br>R Oui<br>S On peut faire un truc [bruit stylo]<br>R Oui oui d'accord ok [bruits papiers]<br>R Donc en fait la surface là c'est un miroir des deux maillages ?<br>En fait. C'est entre les deux maillages ce que tu veux faire ? non ?<br>S Un miroir?<br>R Bah là c'est un peu comme si tu faisais celui là et puis que tu faisais ça.<br>S ah oui |
| Echelle optique                            | [c5] Recherche d'une régularité du motif pour ne pas « casser la forme »   | 10h15 [c5]<br>S non non non on va casser la forme si on fait ça.<br>R bah ça dépend si c'est régulier on va rien casser du tout.  |
|  | → [mp1] induit une OPERATION DE MODELISATION ( ?) une recherche des patterns   | 10h17<br>R Et bah c'est ça que je disais avec mon truc régulier là. Si on fait ça mais régulièrement dans chaque euh... ça casse pas la forme.<br>S oui mais c'est pas un attracteur alors.<br>R non c'est autre chose... C'est pas aléatoire non plus [mp1]  |
|  | Anticipation d'une OPERATION DE MODELISATION liée à l'instanciation, ceci en lien avec l'échelle optique [c5]        | 10h22<br>S ok. On fera pas trop de hauteur pour ce point là sinon ça fera « Bouf »...[rire] Juste subtilement quelque chose...  |
| Op. MP de choix de méthode/géométrie [mp2] | Op. de MP induite par l'échelle optique [c5]   | 10h18<br>[bruits de papiers] en fait il faut diviser la surface en secteurs euh en sections régulière   |
|  | Op. de MP induite par l'échelle de modèle [c4]   | et ensuite on donne une petite normale euh à chaque centre [S répètes « une normale en chaque centre oui»] de surfaces  |
|  |  | et après euh... on voit ce qu'on fait ! [rire]<br>S oui si ça marche vraiment<br>R oui parce qu'il faut un treillis par dessus aussi<br>S (*31'25_10h17) oui il faut deux treillis oui  |
| Op. MP de mise en relation [mp3]           | Mise en relation des intentions du treillis et des éléments existants (la surface)                                   | 10h18 30s<br>R <u>Le premier treillis il part, c'est notre surface</u> , et en dessous on a des normales qui partent vers le bas.   |
| Echelle géométrique                        | [c6] dessin plus précis de la façon dont le treillis « se lève »   | S Oui. Maintenant je vois c'est... en fait c'est un point qui se lève. Attend un carré (bruit de dessin) oui, sur ta surface<br>R oui<br>S et tu la lève chaque fois un point, une ligne comme ça, une ligne comme ça et la surface comme ça. Et après de nouveau, alors il faut lever chaque fois.. voilà..  |



|   |   |   |
|---|---|---|
|   |   |   |
| Op. MP de découpage [mp4]   | Découpage de sous-surfaces en relation à l'échelle géométrique [c6] et l'échelle de modèle [c4] pour y appliquer un morphing  | et après on même faire sur chaque sous-surface comme ça, on peut faire un box, un morphing...   |
| Op. MP de choix de méthode/ programmation [mp5]                                     | Choix d'un pattern qui correspond aux intentions définies par l'échelle optique [c5] et géométrique [c6]  | <p>R Ça va être terrible !</p> <p>On commence par diviser la surface en parties régulières...</p> <p>S euh.. On fait comment... Je crois que se serait le plus facile avec euh...</p> <p>R Comme ça [bruit de tapotement sur la table (probable désignation d'un pattern)] En des sous-domaines avec...</p> <p>S Oui, il y a deux moyens j'ai vu, attend... ici [clic**]</p> <p>R C'est le pattern « créer un réseau de points »</p> <p>S hum... j'ai vu, je crois que j'ai vu... [clic] Comme ça ! « sous-surfaces » et on fait un « explode » après. <u>Comme ça tu as à chaque fois un carré et tu peux après ...</u></p> <p>R Oui, ok</p> |
| Op de MP de choix de méthode/ programmation [mp6]                                   | Le calcul de l'efficacité du modèle tel qu'ils l'ont pensé jusque là, questionne la faisabilité du modèle paramétrique. Cela provoque une nouvelle conception du modèle et de l'appropriation du pattern choisi [mp5] | <p>S hum... attend. Faut bien le penser. Comme ça tu as quatre carrés [dessin 1] Si c'est 1, 0, 2, 3 ; probablement pour le prochain... 0, 2 ... [rire] Ok, <u>ce sera un peu trop difficile. Parce que pour chaque petite surface ce sera un autre point.</u></p> <p>R Oui mais est-ce qu'on ne peut pas faire un ensemble de quatre surfaces à chaque fois ? Est-ce qu'on ne peut pas réunir...</p> <p>S Ah oui ! Et on prend le milieu et hop !</p> <p>R ce sera plus facile...</p> <p>S Evidement</p> <p>R Sinon là on y est demain encore, c'est pas possible ! [rire]</p>   |
| Opération de collaboration  | Proposition d'un « test » de la méthode pour ensuite la généraliser   | <p>S Exactement. Peut être qu'il faut d'abord construire ça à part, bien pensé, et après l'appliquer sur les surfaces.</p> <p>R Euh... oui mais ça bien fait il faut que ça corresponde à notre surface, il faut que ça corresponde à une division de notre surface. Tu vois ce que je veux dire ?</p> <p>S oui...</p>  |
| Echelle géométrique et op. de MP de choix de méthode géométrie/ programmation [mp7] |   | <p>R Je pense qu'il faut d'abord diviser la surface et ensuite on... essaie de les regrouper par 4. Il faut qu'on ait un... Il faut qu'on divise par un multiple de 4 [rire]. Tu vois ce que je veux dire ?</p> <p>S Oui euh ... Oh non, il faut juste que ce soient des carrés.</p> <p>R Oui mais qu'il ne nous reste pas deux surfaces à la fin, toutes pourries qui ne nous servent à rien.</p> <p>Si on divise toute notre surface,</p> <p>S humhum,</p> <p>R en carrés,</p> <p>S oui</p> <p>R et qu'on les prend par 4 pour faire notre...</p>   |

|   |  |  |
|---|--|--|
|   |  | <p>S Ah non non <u>ça c'est une surface</u></p> <p>R <u>ça c'est une seule ?</u></p> <p>S <u>oui oui, et après on divise celui là par 4.</u></p> <p>R D'accord bah c'est plus simple, ok.</p> <p>S Ca sera la galère... [rire]</p> <p>R ok</p> <p>S Je pense juste de d'abord faire ça...euh...</p>  |
| Echelle de niveau de conception               |  | <p>S euh... pour être sûr...</p> <p>R On peut faire ctrl S ?</p> <p>S oui [rire] [clavier] ok bon. On va subdiviser.</p> <p>R C'était quoi « divide »...</p> <p>S « Divide » alors d'abord à l'intègre ...</p> <p>R Diviser en combien ?</p> <p>S pff... je sais pas [murmures]</p> <p>R Mets en « integers » je sais plus quoi...</p> <p>S Oh my god [rire] ok euh... c'est « divide »..</p> <p>R « Domain »</p> <p>S C'est ça... euh... [mots incompréhensibles] surface</p> <p>R Celui là</p> <p>S Non c'est celui là. C'est ça ?</p> <p>R Oui, t'as raison. C'est ce qu'on avait vu.</p> <p>Ok. Alors... ça marche.</p> <p>S On aura besoin de plus. Pour avoir un bon... [mots incompréhensibles]</p> <p>R Peut-être pas le même chiffre. Ok.</p> |
| Instanciation/<br>échelle optique<br>[mp8]    | Choix d'une variable en fonction d'une échelle optique | <p>S ok. On fera pas trop de hauteur pour ce point là sinon ça fera « Bouf »...[rire] Juste subtilement quelque chose...</p>   |
| Echelle de modèle                             |  | <p>S Bon maintenant « explode ».</p> <p>R tu préfères qu'on teste sur une surface ?</p> <p>S Non ça va marcher... Non on va directement faire...</p> <p>Maintenant il faut... euh list item... extraire chaque point. Vertice hop ...[mots incompréhensibles]... ça s'intègre 1 [probable manipulation de slider]</p> <p>Attends je vais sauvegarder</p> <p>On prend comme image euh... de référence [clavier]</p> <p>R Pas ici. Mets le sur le bureau.</p> <p>S J'ai deux bouquins de candela à la maison je suis fan c'est incroyable il fait des portées de 30m avec une épaisseur de 3 cm de béton</p>   |
| Inférence<br>(opération pragmatique)<br>[mp9] | Test de la méthode pour un cas « extrême »             | <p>Ok maintenant il faut... 1.2.3.1... (dessin) Attend parce que j'ai dessiné un carré mais si on fait une surface comme ça, est-ce que c'est encore le milieu ? [dessin 3]</p>  <p>R Oui bien sur</p>   |

|  |   |  |
|--|---|--|
| <p>Op MP choix de méthode programmation/géométrie [mp10]</p> | <p>Modélisation d'une normale à la surface à partir du centre</p> | <p>Oui, un parallélogramme, normalement...</p> <p>il faut prendre la ou les deux touches pour être sur. Et en plus moi je dessine plat mais ça sera courbé</p> <p>R Oui il faut réellement trouver le centre de la surface. De chaque surface.</p> <p>S ok alors on construit une ligne entre 0 et 2, et 1 et 3 sur la surface et on prend le point...</p> <p>R oui</p> <hr/> <p>S euh... ça existe [silence clic] je l'ai utilisé il y a pas longtemps</p> <p>R déjà on peut créer les courbes</p> <p>[silence]</p> <p>S ça j'aime pas trop non... Parce qu'il demande u v</p> <p>R oui</p> <p>[silence]</p> <p>S d'accord. Bon. Alors est-ce qu'on peut faire « eval »... [mots incompréhensibles]. Tu vois le problème ?</p> <p>R Il faut pas qu'on utilise 15 surfaces, c'est pas possible...</p> <p>S Alors attend [mots incompréhensibles]...</p> <p>(silence) bingo</p> <p>(silence) [rire]</p> <p>S Attend mais est-ce que c'est vraiment nécessaire qu'on trouve sur la surface</p> <p>Attend mais y a un moment où dans un exercice on a réussi à faire des normales au centre de chaque surface</p> <p>S Oui « evaluate » surface, oui oui oui hum... c'est quoi ça ? ce sera plutôt ça... normalement ça devrait marcher... trouver la normale au milieu... comment faire ?... il faut faire un point... oui... ah non... Misère</p> <p>R ils viennent d'où tous ces points ?</p> <p>S C'est la merde parce que les... euh... n'importe quoi</p> <p>R Oui</p> <p>S Euh...</p> <p>R ça c'est quoi, là haut, au dessus, non, oui</p> <hr/> <p>S attend, on a une surface [dessin 4] on peut faire peut être euh... (silence appuyé)</p> <p>R C'est des plans là ?</p> <p>S mmh ? oui. Je me demande si on peut pas trouver... mmh je crois qu'on peut dire que... Si on faisait un plan, [dessin 5] on va tromper le truc un peu on fait une surface on fait une sous-surface, on dit que la normale au milieu sera plus ou moins la normale...</p> <p>R Oui on peut</p> <p>S surface... [mots incompréhensibles]... oh non... ça va pas marcher... ! (R souffle)</p> <p>Si je fais 4 points avec un plan ça sera de nouveau courbé</p> <p>R oui enfin ça sera...</p> <p>S plat ! [rire]</p> <p>R Comment on peut faire ?</p> <hr/> <p>S si on fait une autre division sur la surface... il trouve un point ici, un point ici... Si on pouvait faire, il trouve un point la un point la a chaque fois [dessin 6]</p> |
|--|---|--|

## 2.20 Tableau de codage des vidéos « recours au web »

Les premières analyses et les limites des données sont indiquées en fin de tableau pour chaque binôme. Les séquences grisées dans les tableaux indiquent le repérage de moments potentiellement intéressants à analyser de façon plus approfondie.

### A1 Marta et Elena

| <i>tps</i> | <i>Recherche sur le site so823</i>  | <i>Ouverture de fichiers de patterns</i>  | <i>Recherche d'autres références internet</i> |
|------------|---|---|---|
| 11h08      |   | Recours à des fichiers précédents, déjà présents sur leur poste. Recherche fichier « séance 4 »                 |   |
| 11h11      | Ouverture du site des patterns so 821, puis du cours de la séance 4             |   |   |
| 11h15      |   | Copier coller d'un modèle GH construit en cours (spirale)   |   |
| 11h18      | Ouverture du site du cours, séance 3. Puis ouverture d'un fichier nommé seance3 |   |   |
| 11h32      | Site du cours séance 11   |   |   |
| 12h41      | puis site so823 sur séance 03.  | Ouverture d'explorer sur un fichier « city-hall.ghx », puis explorer sur fichier « seance3 exercice 11531.ghx » |   |
| 12h45      | Site so823 sur séance 06  |   |   |
| 12h50      | Site so823 sur séance 03  |   |   |
| 12h55      | Site so823 seance01   |   |   |

→ Informations partielles, échec de la première vidéo (départ traces existante à 10h57)

→ Référence aux cours et aux images des modèles GH, copier-coller d'un seul pattern (spirale) finalement non utilisé

### C1 Toshi et Pierre

| <i>tps</i> | <i>Recherche sur le site so823</i> | <i>Ouverture de fichiers de patterns</i>            | <i>Recherche d'autres références (internet)</i>   |
|------------|------------------------------------|---|---|
| 10h45      |                                    |   | Ouverture d'un fichier précédent (pas fait pendant le cours so 823) Exercice01_btT2.ghx |
| 11h07      |                                    |   | Ouverture du fichier Hexachaos_Co-de-iT_GH080010.ghx                                    |
| 11h08      |                                    | Ouverture fichier Evaluer-planeite-surface.ghx      |   |
| 11h17      |                                    | Ouverture fichier creer-srf-projetee-attracteur.ghx |   |

|       |  |  |                                     |
|-------|--|--|-------------------------------------|
| 11h29 |  |  | Ouverture d'un fichier couple01.ghx |
|-------|--|--|-------------------------------------|

→ Beaucoup de copier-coller de modèles précédents. Appropriation sur le projet : « test » des effets.

→ Bonne manipulation de Grasshopper mais pas de « prise de risque ». Peu de verbalisation => données non utilisables pour une analyse approfondie

## F1 Emiliano et Fabian

| <i>tps</i> | <i>Recherche sur le site so823</i>             | <i>Ouverture de fichiers de patterns</i>  | <i>Recherche d'autres références ( internet)</i>           |
|------------|--|---|--|
| 09h55      | Biblio patterns<br>Cours séance01              |   |  |
| 09h56      | Cours séance 02, Cours séance 03,<br>Séance 06 |   |  |
| 09h57      | Séance 08, Séance 09<br>Biblio patterns        |   |  |
| 09h59      |  | Téléchargement RepoOutofPlane.ghx   |  |
| 10h07      |  | Ouverture du fichier RepoOutofPlane.ghx   |  |
| 10h09      |  | Téléchargement Reaction.attracteur.ghx  |  |
| 10h10      | Cours séance 04, seance 03,                    |   |  |
| 10h11      |  | Ouverture fichier crer maillage.ghx   |  |
| 10h12      |  | Ouverture fichier projeter delaunay.ghx   |  |
| 10h20      | Cours 1, cours2, cours 4                       |   |  |
| 10h21      |  |   | Mathcurve.org  |
| 10h22      | Cours 07                                       |   |  |
| 10h23      |  |   | Google « voronoi architecture »                            |
| 10h24      | Cours 08, intitulé du td,                      |   |  |
| 10h25      |  |   | <a href="http://www.andrew.cmu.edu">www.andrew.cmu.edu</a> |
| 10h26      |  |   | Parametricmodel.com  |
| 10h27      | Biblio patterns                                |   |  |
| 10h28      |  | Créer rf niveaux.ghx  |  |
| 10h31      | Biblio pattern                                 | Téléchargement de creer srf projetée<br>attracteur.ghx  |  |
| 10h32      |  | Ouverture Reaction-reacteur.ghx   |  |
| 10h42      | Biblio p                                       |   |  |
| 11h50      | Biblio p, seance 08, biblio patterns,          | Ouverture de projeter delaunay.ghx  |  |
| 11h53      | biblio   | Téléchargement et copier coller du fichier<br>créer surface projeter random delaunay.ghx                                |  |
| 12h02      |  | Ouverture créer surface niveaux.ghx   |  |
| 12h05      | biblio   | Telechargement et copier coller créer crv<br>voronoi.ghx<br>Téléchargement/ouverture d'autres patterns<br>(pas lisible) |  |
| 12h15      | Cours 03, biblio... (pas lisible)              |   |  |
| 12h44      | Seance 07                                      |   |  |

→ Etudiants un peu perdus. Navigation sur le cours à la recherche de quelque chose.

→ Aucune réalisation, déception des étudiants, ils déclarent lors de discussions avec l'enseignant au cours de la passation : « les patterns ne sont pas adaptés à ce qu'on veut faire » « on ne trouve pas de patterns intéressants ».

---

**A2** Rebecca et Sebastiaan

| <i>tps</i> | <i>Ouverture d'un site de patterns</i> | <i>Ouverture de fichiers de patterns</i> | <i>Recherche d'autres références internet</i>   |
|------------|--|--|---|
| 10h13      | patterns so823                         |  |   |
| 10h16      |  |  | google « felix candela »                        |
| 10h19      | so823 cours séance 2                   |  |   |
| 11h42      | so 823 cours séance 3 (morphing)       |  |   |
| 11h44      |  |  | google « felix candela » puis page web reinhaus |
| 12h17      | cours séance 3 (morphing)              |  |   |

→ Résultats biaisés car les étudiants utilise également une impression papier de la liste des patterns

→ Référence aux cours et aux images des modèles Grasshopper mais pas de téléchargement de patterns : les patterns sont reconstruits

---

**B2** Julia et Maxime

| <i>tps</i>                  | <i>Recherche sur le site so823</i>                      | <i>Ouverture de fichiers de patterns</i>                 | <i>Recherche d'autres références internet</i> |
|-----------------------------|---|--|---|
| 09h53                       | Biblio patterns   |  |   |
| 09h57                       |   | Ouverture d'un fichier creer_srf-projetee_attracteur.ghx |   |
| 09h58                       | Biblio patterns   |  |   |
| 10h05                       | Puis ouverture du cours so823 séance 06                 | Retour sur le fichier creer_srf-projetee_attracteur.ghx  |   |
| 10h05<br>à ...              | Alternance manipulation GH et retour sur cours seance06 |  |   |
| 10h23                       | Biblio so823  |  |   |
| 10h25                       |   | Téléchargement du fichier creer-srf-niveaux.ghx          |   |
| 10h28                       | Biblio so823  |  |   |
| 10h29                       | Cours so823 séance 08                                   |  |   |
| 10h36                       | Biblio so823  |  |   |
| Manque 1h30 (reprise à 12h) |   |  |   |
| 12h02                       | Biblio so823  | Téléchargement du fichier creer-srf_meth_shell.ghx       |   |
| 12h04                       | Biblio so823  | Téléchargement du fichier creer-motif-morphing.ghx       |   |

→ Répétition d'une procédure : Exploration de la bibliothèque de patterns, téléchargement, ouverture du fichier, copier coller sur le fichier de travail, manipulation, puis suppression si cela ne convient pas

---

**C2** Amanda et Max

| <i>tps</i> | <i>Recherche sur le site so823</i> | <i>Ouverture de fichiers de patterns</i>  | <i>Recherche d'autres références internet</i> |
|------------|------------------------------------|---|---|
| 10h06      |                                    | Fichier 06194s4exo1b.ghx (seance1, city hall)   |   |
| 10h12      | Séance 12 intitulé du TD           |   |   |
| 10h16      |                                    | Fichier 06194s4exo8 correction.ghx (waterloo)<br>(Collé sur le fichier de travail à 11h23)  |   |
| 11h23      |                                    | début manipulation/appropriation exercice<br>waterloo   |   |
| 11h30      |                                    | Retour sur le fichier 06194s4exo8 correction.ghx<br>original  |   |
| 11h34      |                                    | Ouverture du fichier 04025s4ex5version1.ghx<br>Ouverture du fichier 04025s7ex7-delaunay.ghx<br>Ouverture du fichier 04025s11teste1erexo.ghx |   |
| 11h35      |                                    | Ouverture du fichier 04025s10ex101etage.ghx   |   |
| 11h36      |                                    | Retour sur 04025s11teste1erexo.ghx  |   |
| 11h42      |                                    | Ouverture du fichier 04025s4exo1b.ghx<br>(copier coller à 11h44)  |   |
| 11h46      | Biblio patterns                    | Téléchargement de creer-motif-morphing.ghx<br>Creer-motif-maillage.ghx (copier coller à 11h52)  |   |
| 11h57      |                                    | Ouverture du fichier pattern1.ghx   |   |
| 12h02      |                                    | Copier coller crer-surface-projetee-attracteur.ghx  |   |

→ Grosses difficultés à manipuler Grasshopper et à comprendre comment marchent les patterns (à vérifier avec l'audio)

## **D2** Alessandro et Veronika

| <i>tps</i> | <i>Recherche sur le site so823</i> | <i>Ouverture de fichiers de patterns</i>  | <i>Recherche d'autres références ( internet)</i> |
|------------|------------------------------------|---|--|
| 09h58      | Séance 8, puis intitulé TD         |   |  |
| 09h59      |                                    |   | Kd3 surf   |
| 10h03      | Seance courbes par fonctions       |   |  |
| 10h05      | Biblio patterns                    |   |  |
| 10h06      |                                    | Creer-crv-math-spirale-3d.ghx (ouverture +<br>manipulation)                             |  |
| 10h17      |                                    | Pattern-1.ghx (ouverture) (groupes surface<br>projetee attracteur et creer motif croix) | Dico français-allemand en<br>ligne               |
| 10h31      |                                    | Ouverture pattern-2 con mesh.ghx  |  |
| 10h56      | Biblio patterns                    |   |  |
|            |                                    | Créer-motif reactif attracteur.ghx  |  |
| 11h11      |                                    | Pattern1.ghx  |  |

→ Manque 1h de vidéo environ

→ Beaucoup d'exploration et de réutilisation de patterns, peu (pas) de vérification sur les cours.

→ Beaucoup de difficulté à contrôler le modèle

## **A3** Paul et Ludo

| <i>tps</i> | <i>Recherche sur le site so823</i> | <i>Ouverture de fichiers ( de patterns)</i> | <i>Recherche d'autres références (</i> |
|------------|------------------------------------|---|--|
|------------|------------------------------------|---|--|

|  |                                    |   |  |
|--|------------------------------------|---|--|
|  |                                    |   | <i>internet</i>  |
| <b>J1</b> (9h36-11h58) Modélisation de la couverture, du bassin. Début d'exploration pour « terrasse » |                                    |   |  |
| 10h29  | Téléchargement d'anciens exercices |   |  |
| 10h32  |                                    | Ouverture du fichier perot13oct.ghx (exercice paroi salle d'expo) et pero17nov.ghx<br>Copier coller dans fichier de travail |  |
| 10h42  |                                    | Récupération d'un fichier d'exercice précédent (salle d'exposition) comprenant un pattern co-de-it détourné                 |  |
| 11h13  |                                    | Ouverture du fichier midpoint.ghx (dans explorer clef usb)  |  |
| <b>J2</b> (09h10 – 12h00) modélisation des terrasse-cabine de plage                                    |                                    |   |  |
| 09h20  |                                    |   | Design patterns GH (andrew)  |
| 09h31  |                                    |   | Co de it, téléchargement de Panels dispatch_Co-de-iT   |
| 09h36  |                                    | Ouverture de dispatch_Co-de-iT  |  |
| 09h39  |                                    |   | Parametric model   |
| 10h18  |                                    |   | Parametric model telechargement de « compare every item in a list » (ouverture et exploration) |
| 10h25  |                                    | Copier coller de dispatch_Co-de-iT dans le modèle du TD<br>Explorations, transformations pour appropriation                 |  |
| 10h50  |                                    | Retour/observation du fichier dispatch_Co-de-iT original  |  |
| 11h12  |                                    |   | Parametric model   |

- Bonne exploration/appropriation des patterns
- Bonne maîtrise de Grasshopper, bonne organisation du modèle
- Réutilisation facile des anciens fichiers de l'étudiant Paul
- Fort déséquilibre entre les deux étudiants du binôme, projet entièrement mené par Paul)

### **B3** (Alice et Antoine)

#### **Poste1**

| <i>tps</i>              | <i>Recherche sur le site so823</i> | <i>Ouverture de fichiers ( de patterns)</i>  | <i>Recherche d'autres références ( internet)</i> |
|-------------------------|------------------------------------|--|--|
| <b>J1</b> (09h29-11h16) |                                    |  |  |
| 09h41                   |                                    | Ouverture du fichier cours1.ghx (city hall)  |  |
| 09h42                   |                                    | Ouverture du fichier cours4.ghx (waterloo) exploration   |  |
| 10h10                   |                                    | Cours4   |  |
| 10h34                   |                                    | Ouverture du cours 8 puis copié collé sur le fichier final et appropriation (motif sur surface)<br>Appliquer puis transformé |  |



|                   |   |  |  |
|-------------------|---|--|--|
| 11h14             | Biblio, telechargement créer motif reactif attracteur |  |  |
| <b>J2 (09h17)</b> |   |  |  |
| 09h57             |   | Motif reactif attracteur.ghx   |  |
| 10h00             | Biblio téléchargement de créer motif maillage.ghx     |  |  |
|                   |   | Ouverture/exploration du cours5.ghx (waterloo)   |  |
| 10h13             |   | Copier coller d'un pattern ( ?) (avec scripting) exploration, application à une des surfaces |  |
| 10h22             | Téléchargement créer motif morphing.ghx               |  |  |

### Poste 2

| <i>tps</i>         | <i>Recherche sur le site so823</i>  | <i>Ouverture de fichiers ( de patterns)</i> | <i>Recherche d'autres références ( internet)</i> |
|--------------------|---|---|--|
| <b>J2 (09h18-)</b> |   |   |  |
| 09h24              | Ouverture de la page biblio   |   |  |
| 09h48              | Exploration de la page biblio   |   |  |
| 09h55              | Récupération d'anciens exercices  |   |  |
| 10h19              | biblio  |   |  |
| 10h47              | Ouverture du cours de la seance 3 (morphing) puis seance motif attracteur |   |  |

→ Utilisation de deux postes : un pour modéliser, l'autre pour des taches annexes (faire des mesures le jour 1 ; faire des recherches internet le jour 2)

→ Observation du fichier cours4.ghx pour le **reproduire** (une surface divisée par des plans verticaux)

→ Difficultés dans la réutilisation de patterns dans des contextes autres que vus en cours

### C3 (Pierre Luc et Martial)

| <i>tps</i>                           | <i>Recherche sur le site so823</i> | <i>Ouverture de fichiers ( de patterns)</i>                | <i>Recherche d'autres références ( internet)</i>                       |
|--------------------------------------|------------------------------------|--|--|
| <b>J1 _Poste 1 (9h41-)</b>           |                                    |  |  |
|                                      |                                    | Ouverture de « Hexagonal field Co-de-It »                  | (probable exploration sur co-de-it sur poste 2)                        |
|                                      |                                    | Ouverture seance11martialmarquet.ghx                       |  |
| 10h21                                |                                    | exploration-adaptation – debugging de “hexagonal field”    |  |
| 11h40                                |                                    | Ouverture d'un pattern meshtobrep.ghx                      | (Probable exploration sur haumontdesigntech.wordpress.com sur poste 2) |
| <b>J2_Poste Martial (9h22-12h10)</b> |                                    |  |  |
|                                      |                                    | Exploration-modification du modèle de la séance précédente |  |
| 10h04                                |                                    |  | Ouverture du site martialmarquet.fr                                    |
| 10h39                                |                                    |  | Ouverture du site martialmarquet.fr                                    |

|   |                          |  |  |
|---|--------------------------|--|--|
|   |                          |  | projet « Rise Above II »                                   |
| 12h05                                     |                          |  | Ouverture 3D Warehouse pour chercher modèle de personnages |
| <b>J2</b> _Poste Pierre-Luc (10h23-12h48) |                          |  |  |
| 10h31                                     | Ouverture cours Séance 8 |  |  |
| 10h32                                     | Ouverture cours Séance 9 |  |  |
| 11h03                                     |                          |  | Ouverture google image recherche « flagey arret tram »     |

→ Manque l'enregistrement du poste 2 pendant la 1<sup>ère</sup> séance

→ Difficulté à transformer un pattern (des hexagones à une surface) intervention de l'encadrant pour orienter le travail vers un modèle « faisable » avec le pattern, c'est-à-dire basé sur les lignes pour créer des structure plutôt que sur des surfaces ou mesh difficile à maîtriser.

### **D3** (Charlie)

| <i>tps</i>              | <i>Recherche sur le site so823</i> | <i>Ouverture de fichiers ( de patterns)</i>                 | <i>Recherche d'autres références ( internet)</i>         |
|-------------------------|------------------------------------|---|--|
| <b>J2</b> (19h20-23h21) |                                    |   |  |
| 19h33                   |                                    |   | Ouverture du Rhino4 .NET SDK                             |
|                         |                                    | Ouverture de l'exercice personnel CrvDiscrete.ghx           |  |
| 20h22                   |                                    |   | Ouverture de la page de Marco Sosa sur grasshopper3d.com |
| 22h53                   |                                    | Ouverture du pattern créer motif croix.ghx et copier coller |  |

→ Manque la première séance de travail

→ Paraît comporter un intéressant travail de modélisation paramétrique, mais comprend peu d'information sur l'usage des patterns.

### **E3** (Adrien et Franklin)

| <i>tps</i>                       | <i>Recherche sur le site so823</i> | <i>Ouverture de fichiers ( de patterns)</i> | <i>Recherche d'autres références ( internet)</i>                                  |
|----------------------------------|------------------------------------|---|---|
| <b>J1</b> _poste1 (9h33-)        |                                    |   |   |
| 9h35                             | Ouverture cours seance 9           |   |   |
|                                  |                                    | Ouverture fichier perso 08101cours8         |   |
| 11h14                            | Ouverture de la biblio             |   |   |
| 11h15                            |                                    | Copier coller créer motif maillage          |   |
| <b>J2</b> _Poste 2 (10h12-11h52) |                                    |   |   |
| 10h12                            |                                    |   | Ouverture de parametricmodel, giulio piacentino, co de it, grasshopper ressources |
| 11h52                            | Bibliotheque de patterns           |   |   |
| <b>J2</b> _poste 1 (9h07-11h50)  |                                    |   |   |
| 9h28                             |                                    |   | Ouverture I eat bugs for breakfast, giulio Piacentino, andrew cmu                 |
| 9h32                             |                                    |   | Recherché goggle "pattern grasshopper 3d"   |

|       |  |                                |  |
|-------|--|--------------------------------|--|
| 10h   |  |                                | Ouverture co-de-it, telechargement<br>curvature analysis pattern |
| 10h30 |  | Copier coller pattern co-de-it |  |
| 11h   |  |                                | Telechargement vector deformation<br>co de it et panel dispatch  |
|       |  | Copier coller pattern co-de-it |  |

- Un début de modélisation sur Rhino
- Des difficultés à maîtriser les surfaces et les booléens

---

### F3 (JB et Jérémie)

| <i>tps</i>                      | <i>Recherche sur le site so823</i>   | <i>Ouverture de fichiers ( de patterns)</i> | <i>Recherche d'autres références ( internet)</i> |
|---------------------------------|--|---|--|
| <b>J1_</b> poste 1 (9h34-12h13) |  |   |  |
| 9h41                            | Ouverture de la page des patterns<br>Téléchargement du pattern<br>« heliodon » |   |  |
| 9h44                            |  | Ouverture du pattern « heliodon »           |  |
| 11h52                           |  | Copie du pattern heliodon                   |  |
| <b>J2_</b> poste 1 (9h07-12h13) |  |   |  |
|                                 | Néant  |   |  |

- Manque la fin de la vidéo poste 1 jour 1
- Peu d'utilisation de patterns. Probable précédente pratique experte de Grasshopper.

---

### G3 (Loic et Antoine)

| <i>tps</i>                     | <i>Recherche sur le site so823</i>                    | <i>Ouverture de fichiers ( de patterns)</i>   | <i>Recherche d'autres références ( internet)</i>   |
|--------------------------------|---|---|--|
| <b>J1_</b> poste 1 (10h03-12h) |   |   |  |
|                                |   | Travail sur un fichier<br>« attracteur » (probable fichier<br>perso construit pendant le cours) |  |
| 10h14                          |   | Ouverture d'un fichier perso<br>« city_Hall » correspondant à la<br>séance 2                    |  |
| 10h46                          |   | Ouverture d'un fichier perso<br>« VB02 » probablement construit<br>en cours                     |  |
| 11h05                          |   | Copie de « VB02 » dans le projet,<br>manipulation du code (echec)                               |  |
| <b>J2_</b> poste 2 (10h09-)    |   |   |  |
| 10h35                          | A partir de la page « ressource » du<br>site du cours |   | Ouverture de complexitys.com ;<br>digital morphogenesis, grasshopper<br>ressources, andrew design patterns |
| 11h08                          |   |   | Ouverture de parametric model, co-<br>de-it<br>Telechargement de « panel-dispatch-<br>codeit »             |

|                                |   |   |   |
|--------------------------------|---|---|---|
| 11h14                          |   | Ouverture de panels_dispatch                          |   |
| 11h17                          |   |   | Retour sur le site co-de-it   |
| 11h58                          | Ouverture du cours de la séance 4,<br>puis de la séance 2         |   |   |
| <b>J2_ Poste 1 (10h10-12h)</b> |   |   |   |
| 10h33                          | Ouverture cours séance 2<br>(reproduction du modèle en parallèle) |   |   |
| 11h11                          | A partir du site de ressources                                    |   | Ouverture de parametric model,<br>téléchargement de « heagrid on<br>surface » |
| 11h13                          |   | Ouverture de « heagrid on<br>surface » et exploration |   |
| 11h17                          |   |   | Ouverture de co-de-it,<br>téléchargement de « panels<br>dispatch »            |

→ Manque la première partie de la vidéo du jour 1

→ Contrôle de la modélisation/conception au travers d'une instance sur Rhino → observation des bugs (11h38) => modification de l'intention (11h57)

### H3 (Clément et Simon)

| <i>tps</i>                                  | <i>Recherche sur le site so823</i>   | <i>Ouverture de fichiers ( de<br/>patterns)</i>  | <i>Recherche d'autres références ( internet)</i>                              |
|---|--|--|---|
| <b>J2_ poste « structure » (9h05-12h20)</b> |  |  |   |
| 9h09  | Ouverture du cours 2, du cours 8<br>(VBligne coupée)<br>Séance 6, séance 3, seance 2 |  |   |
| 9h53  | Séance 2   |  | Hotmail (échange de fichiers avec le<br>co-équipier)                          |
| <b>J2_ poste Clément (09h04-12h21)</b>      |  |  |   |
| 09h25                                       |  |  | Ouverture de tegrity.gatech.edu →<br>cours en ligne (échec, pb de<br>pluggin) |
| 9h44  |  | Réutilisation d'un code VB<br>développé en cours |   |
| 9h50  | Séance 8, séance 5, séance 4, séance3,<br>séance 2                                   |  |   |
| 9h54  | Séance 2   |  |   |
| 10h19                                       | Séance 3   |  |   |
| 11h54                                       |  |  | Gmail échange de fichier avec co-<br>équipier                                 |

→Manque toutes les vidéos du jour 1

→Utilisation de rhino pour le contrôle

## 2.21 Codage des samples utilisés par projet

|           | Heliodon |  | Motif-attracteur | Morphing | Créer_maillage | Dispatch-panels | Hexagone |  | creer_crv_math-spirale3D | creer_crv_voronoi |  | Créer_srf_projetee_attracteur | OCOPWave | Commentaires  |
|-----------|----------|--|------------------|----------|----------------|-----------------|----------|--|--------------------------|-------------------|--|-------------------------------|----------|---|
| <b>A1</b> |          |  |                  | 1        |                |                 |          |  |                          |                   |  | 1                             |          | Difficultés à modéliser, utilisation « scolaire » des patterns  |
| <b>B1</b> |          |  |                  |          |                |                 |          |  |                          |                   |  |                               |          | Grande maîtrise de GH, pas de recours aux patterns (probable exploration du pattern attracteur, mais non retenu)                                |
| <b>C1</b> |          |  |                  |          | 1              | 1               |          |  |                          |                   |  |                               |          | Maîtrise de Grasshopper, utilisation rapide des patterns  |
| <b>D1</b> |          |  |                  |          |                |                 |          |  |                          | 1                 |  |                               |          | Grande maîtrise de Generative Components, peu de recours aux patterns (probable abandon du pattern créer surface-section après exploration)     |
| <b>E1</b> |          |  |                  |          |                |                 |          |  | 1                        |                   |  |                               |          | Difficultés à modéliser, utilisation « scolaire » des patterns et des notions du cours  |
| <b>F1</b> |          |  |                  |          |                |                 |          |  |                          | 1                 |  |                               |          | Difficultés à modéliser   |
|           |          |  |                  |          |                |                 |          |  |                          |                   |  |                               |          |   |
| <b>A2</b> |          |  |                  | 1        | 1              |                 |          |  |                          |                   |  |                               |          | Peu d'utilisation des patterns, pas de copier coller mais travail de modélisation plus personnel  |
| <b>B2</b> |          |  |                  | 1        | 1              |                 |          |  |                          |                   |  | 1                             |          | Difficultés à modéliser, les patterns semblent avoir été une aide, mais utilisation « scolaire »  |
| <b>C2</b> |          |  |                  | 1        | 1              |                 |          |  |                          |                   |  |                               |          | Difficultés à modéliser, mais utilisation des patterns  |
| <b>D2</b> |          |  | 1                |          | 1              |                 |          |  | 1                        |                   |  |                               | 1        | Beaucoup d'explorations de patterns   |
| <b>E2</b> |          |  |                  |          |                |                 |          |  |                          |                   |  |                               |          |   |
|           |          |  |                  |          |                |                 |          |  |                          |                   |  |                               |          |   |
| <b>A3</b> | 1        |  |                  |          |                | 1               |          |  |                          |                   |  |                               |          | Intéressante appropriation du pattern de co-de-it mais echec des cabines (dommage)  |
| <b>B3</b> |          |  | 1                |          |                |                 |          |  |                          |                   |  |                               |          | Modèle bien organisé. Idée de départ intéressante, réalisation faible   |
| <b>C3</b> |          |  |                  |          |                |                 | 1        |  |                          |                   |  |                               |          | -intéressante appropriation du pattern co-de-it   |
| <b>D3</b> |          |  |                  |          |                |                 |          |  |                          |                   |  |                               |          | - très bonne maîtrise. Bonne organisation du modèle.<br>-Résultat un peu pauvre mais parfaitement maîtrisé.                                     |
| <b>E3</b> |          |  |                  |          | 1              | 1               |          |  |                          |                   |  |                               |          | - bon travail d'instanciation (choix de la définition de points en fonction du site)<br>- utilisation un peu basique, modèle parfois créé « par |

|  |   |  |   |   |   |   |   |  |   |  |  |   |  |   |
|--|---|--|---|---|---|---|---|--|---|--|--|---|--|---|
|  |   |  |   |   |   |   |   |  |   |  |  |   |  | défaut » (pb tech)  |
| <b><u>F3</u></b>   | 1 |  |   |   |   |   |   |  |   |  |  |   |  | Bonne maîtrise de GH  |
| <b><u>G3</u></b>   |   |  |   |   | 1 | 1 |   |  |   |  |  | 1 |  | - bonne organisation du modèle<br>- utilisation d'éléments dans Rhino<br>- patterns simples mais bien maîtrisés.              |
| <b>H3</b>  |   |  |   |   | 1 |   | 1 |  |   |  |  |   |  | -gros travail d'instanciation (définition de points)<br>-intéressante mise en relation des motifs en fonction de la structure |
|  |   |  |   |   |   |   |   |  |   |  |  |   |  |   |
| <b><u>A4</u></b>   |   |  | 1 |   |   | 1 | 1 |  |   |  |  |   |  | Bonne maîtrise de Grasshopper pour le projet  |
| <b>B4</b>  |   |  | 1 |   |   | 1 |   |  |   |  |  |   |  | Problème de fichier   |
| <b>C4</b>  | 1 |  |   |   | 1 | 1 |   |  |   |  |  |   |  | Intéressant projet, Bonne maîtrise du code  |
| <b><u>D4</u></b>   | 1 |  |   |   |   | 1 |   |  | 1 |  |  | 1 |  | Rapport au site moyen<br>très riche appropriation des codes fournis   |
| <b><u>E4</u></b>   |   |  | 1 | 1 |   | 1 |   |  |   |  |  |   |  | Bel effort d'appropriation des patterns. Bien mené, en particulier l'instanciation  |
| <b>F4</b>  |   |  |   |   |   | 1 |   |  |   |  |  |   |  | Bonne maîtrise de Grasshopper pour le projet, pas de réutilisation intéressante de sample                                     |
| <b><u>G4</u></b>   |   |  | 1 |   |   | 1 |   |  |   |  |  |   |  | Projet assez pauvre sauf très bonne appropriation du sample obligatoire : transformation avec attracteur                      |
| <b><u>H4</u></b>   |   |  | 1 |   |   | 1 |   |  | 1 |  |  |   |  | Utilise beaucoup de supports rhino, Bon travail   |
| Légende :  |   |  |   |   |   |   |   |  |   |  |  |   |  |   |
| En <u>souligné</u> les binômes considérés comme maîtrisant relativement bien Grasshopper |   |  |   |   |   |   |   |  |   |  |  |   |  |   |

## ANNEXE 3 :

# CATALOGUES DE SAMPLES

## Analyse des *samples*, de leurs constructions et de leurs évolutions au regard des expérimentations effectuées

Pour plus de lisibilité, les noms des samples sont indiqués en police « Arial » et les noms des patterns en police « Verdana ».

### 3 Organiser des arbres de données

#### 3.1 *Extrait\_1item\_liste4*<sup>117</sup>

##### Spécificité du sample

Le code `Extrait_1item_liste4` permet à partir d'un arbre de données regroupant des listes de quatre items chacune d'extraire quatre listes. Les listes obtenues regroupent les mêmes items entre eux : tous les items en 0, puis en 1, *etc.* Ce code est à la fois général et spécifique : général parce qu'il peut réorganiser n'importe quel arbre de donnée structuré avec quatre items par branche, mais il est cependant spécifique car il est particulièrement adapté pour assister l'organisation de listes de sommets de sous-surface. L'organisation de ces points permet ensuite de facilement coder un « motif » sur ceux-ci. Ce code se retrouve par ailleurs dans le sample `creer_motif-maillage`.

Ce sample peut être associé au pattern `Place Holder` de R. Woodbury. Dans le pattern `Place Holder` un objet « proxy » est utilisé pour lier deux parties d'un modèle : l'une générant un contexte sur lequel est répété l'autre (Woodbury 2010, p.218). Le code `Extrait_1item_liste4` peut être utilisé comme un fragment d'exemple de pattern `Place Holder` car il permet de réorganiser des listes de points créées à partir d'un objet proxy « sous-surface » (étape 2 Figure 170) pour instancier un composant comme dans le sample `creer_motif-maillage` (étape 4 Figure 170).

---

<sup>117</sup> Ici les *samples* sont nommés par le dernier nom qui leur a été attribué : celui-ci peut avoir changé au fur et à mesure de la construction du catalogue.

Evolution du sample et de son inscription dans les catalogues proposés

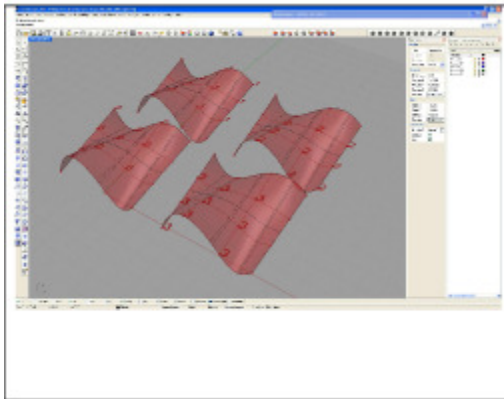
|   |  |   |   |
|---|--|---|---|
|  | <p><b>extrait_1item_liste4</b></p> <p>A partir d'un arbre de données regroupant des listes de 4 items (typiquement une surface divisée en sous surfaces) ce pattern extrait 4 nouvelles listes. Ces listes regroupent les même items entre eux : tous les 0, tous les 1, tous les 2, tous les 3.</p> <p>=&gt; utilise le pattern creer_liste-pts_sous-srf_UV<br/>=&gt;est utilisé dans le pattern creer_motif_maillage</p> | <p><i>Inputs :</i></p> <ul style="list-style-type: none"> <li>- 1 Arbre de données (0; 0 ; 0)</li> </ul> <p><i>Output :</i></p> <ul style="list-style-type: none"> <li>- 4 Listes « à plat »</li> </ul> | <p>Cours so823 séance 2 et 11</p> <p>extrait_1pt_liste4.ghx</p> |
|---|--|---|---|

Figure 170 : Code Extrait\_1item\_liste4 (catalogue So823-2011)

Dans la version So823-2012, le code Extrait\_1item\_liste4 a été modifié par rapport à la version so823-2011 : des composants différents (plus efficaces et rendant le code plus concis) ont été utilisés. Cependant les entrées et les sorties sont les mêmes dans ces deux versions du codes. Des exemples ont été ajoutés dans le catalogue so823-2012 pour exemplifier l'utilisation du sample : - creer\_liste-sous-surface et creer\_motif-maillage.

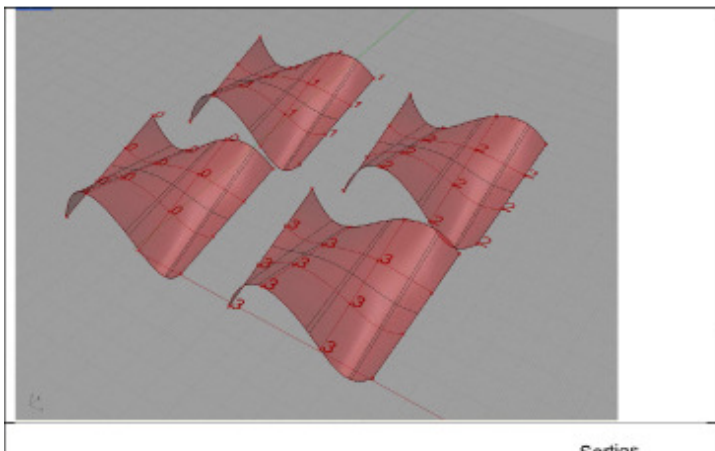
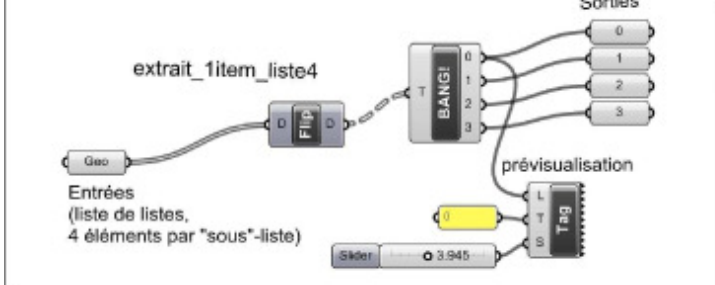
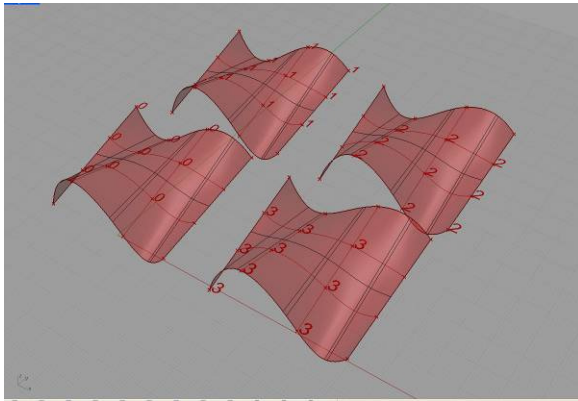
|  |  |  |
|--|--|--|
|   | <p><b>extrait_1item_liste4</b></p> <p>→ Extrait les items d'une liste de 4 objets</p> <p>Inputs :<br/>Géométries quelconques en liste de 4 items</p> <p>Outputs :<br/>4 listes spécifiques</p> <p>Cf. SEANCE 5</p> | <p>GH v0.8.0006</p> <p><b>extrait_1item_liste4.ghx</b></p>   |
|  <p>Entrées (liste de listes, 4 éléments par "sous"-liste)</p> <p>Sorties</p> <p>prévisualisation</p> | <p>Exemples :</p> <p>Extraire des sommets de sous surface (pour motifs...)</p>   | <p>GH v0.8.0006</p> <p><b>creer_liste-pts_sous-srf_UV.ghx</b></p> <p>GH v0.8.0006</p> <p><b>creer_motif_maillage.ghx</b></p> |

Figure 171 : Code Extrait\_1item\_liste4 (catalogue So823-2012)

Sample et inscription retenue dans le catalogue DNArchi-2012





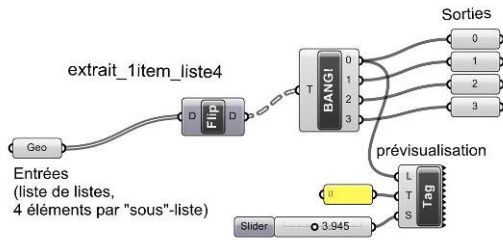
### extrait\_1item\_liste4

→ Extrait les items  
d'une liste de 4 objets

Inputs :  
Géométries  
quelconques en liste  
de 4 items

Outputs :  
4 listes spécifiques

GH v0.8.0006  
[extrait\\_1item\\_li  
ste4.ghx](#)



Cf. Articles  
« Grasshopper et les  
arbres de données,  
introduction »  
« Le pattern Place  
Holder »

Place Holder    Arbres de données  
Motifs    « Set »

Exemples :  
Extraire des sommets  
de sous surface (pour  
motifs...)

GH v0.8.0006  
[creer\\_liste-  
pts\\_sous-  
srf\\_UV.ghx](#)

GH v0.8.0006  
[creer\\_motif\\_ma  
illage.ghx](#)

Figure 172 : Code Extrait\_1item\_liste4 (catalogue DNArchi-2012)

Le code Extrait\_1item\_liste4 retenu pour le catalogue DNArchi est proche de celui construit pour le catalogue So823-2012. Des modifications ont été effectuées quant aux notions permettant de comprendre le code : un important article sur les arbres de données y est associé *via* un lien hypertexte.

Les mots clefs choisis pour caractériser le sample Extrait\_1item\_liste4 sont :

- relatifs à la culture : Motif (+ « pattern » en langage courant ?)
- relatifs aux composants et fonctions utilisés : Set
- relatifs aux notions sollicitées : Arbres de données
- relatifs aux patterns impliqués : Place Holder
- relatifs aux échelles dominant l'usage du : sans

sample :

### 3.2 Extrait\_valeur-max-min

#### Spécificité du sample

Le sample `extrait_valeur-max-min` permet de trier une liste de valeurs et d'en extraire le(s) extrême(s). Il utilise des composants de manipulation d'arbres de données. L'usage de ce code a été observé dans la pratique de l'agence Decode Bim en particulier pour pouvoir retrouver des items à partir d'éléments de la scène Rhino. L'agence Decode travaillant pour le projet Canopée les Halles sur des fichiers Rhino comprenant des surfaces de références (générées par eux-mêmes ou par les architectes), ils sont souvent confrontés à la question de l'automatisation de la sélection des entrées de leurs modèles. Ce sample était souvent utilisé, par exemple dans des codes visant à sélectionner les éléments les plus proches d'un objet de références, les surfaces aux aires les plus élevées, *etc.* Ce sample est donc un fragment pouvant être associé au pattern **Selector** de R. Woodbury : pattern qui propose de sélectionner des objets ayant des propriétés spécifiques (Woodbury 2010, p.245). Des exemples appliquant plus directement ce pattern sont proposés dans le catalogue `so823-2012`.

Evolution du sample et de son inscription dans les catalogues proposés

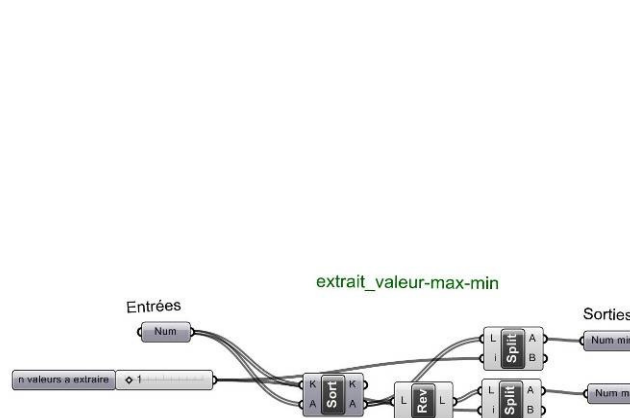
|  |  |   |
|--|--|---|
|  | <p><b>extrait_valeur-max-min</b></p> <p>→ Extrait la ou les valeur(s) max et min d'une liste</p> <p>Inputs :<br/>Liste (A PLAT) de valeurs</p> <p>Outputs :<br/>valeur(s) max et min</p> <p>Cf. SEANCE 5</p> | <p>GH v0.8.0006<br/><b>extrait_valeur-min-max.ghx</b></p>   |
|  | <p><b>Exemples :</b></p> <p>// Extraire les surfaces aux aires min ou max depuis une liste</p> <p>// Sélectionner le(s) objet(s) par rapport à leur proximité à un autre objet</p>                           | <p>GH v0.8.0006<br/><b>extrait_srf-max.ghx</b></p> <p>GH v0.8.0006<br/><b>selectionner_pts_prox.ghx</b></p> |

Figure 173 : Extrait\_valeur-max-min (catalogue So823-2012)

Le sample `extrait_valeur-max-min` a été ajouté à la version `so823-2012`. Les exemples de ce code donnés dans le catalogue sont :

- `Extrait_srf-max` : ce code permet d'extraire dans une liste de surface, celles qui ont les aires maximum et minimum. Ce modèle correspond au pattern **Selector** puisqu'il vise bien à sélectionner des objets en fonction de leurs caractéristiques : ici l'aire minimum ou maximum.

- `selectionner_pts_prox` : ce code permet de sélectionner n points en fonction de leurs proximités à un point de référence. Comme l'exemple précédent, ce code peut être assimilé au pattern **Selector**.



**Selector Arbres de données**

« **Sort List** »

### extrait\_valeur-max-min

→ Extrait la ou les valeur(s) max et min d'une liste

Inputs :

Liste (A PLAT) de valeurs

GH v0.8.0006  
**extrait\_valeur-min-max.ghx**

Outputs :

valeur(s) max et min

Cf. Articles

« Grasshopper et les arbres de données, introduction »

« Le Pattern Selector »

Exemples :

// Extraire les surfaces aux aires min ou max depuis une liste

GH v0.8.0006  
**extrait\_srf-max.ghx**

// Sélectionner le(s) objet(s) par rapport à leur proximité à un autre objet

GH v0.8.0006  
**selectionner\_pts\_prox.ghx**

**Figure 174 : Extrait\_valeur-max-min (catalogue DNArchi-2012)**

Le code retenu pour le catalogue DNArchi-2012 est le même que celui développé pour le catalogue so823-2012. Il comprend les mêmes exemples. Il intègre cependant en plus un lien hypertexte vers un article relatif aux arbres de données sur Grasshopper.

**Les mots clefs choisis pour caractériser le sample extrait\_valeur-max-min sont :**

- relatifs à la culture : sans
- relatifs aux composants et fonctions utilisés : « sort List »
- relatifs aux notions sollicitées : Arbre de données
- relatifs aux patterns impliqués : Selector
- relatifs aux échelles dominant l'usage du sample : sans

### 3.3 mapping\_list

#### Spécificité du sample

Le sample `mapping_list` permet de « mapper » une liste de valeur vers un domaine ciblé. Ce principe s'appuie principalement sur le composant Grasshopper « ReMap » mais il est ici proposé déjà modélisé. Ce sample correspond au pattern `mapping` de R. Woodbury, ce pattern « *use a fonction in a new domain of range* » (Woodbury 2010, p.252).

Ce code a été conçu lors d'un travail avec des étudiants de Marnes-la-Vallée. Dans ce projet, ce sample a été développé en lien avec des opérations de conception relevant de l'échelle de modèle, de l'échelle fonctionnelle, et de l'échelle de visibilité. Cependant, les liens entre ce sample et ces trois classes d'opérations de conception semblent circonstanciés et liés aux spécificités du projet. Il ne semble pas que l'usage de ce sample soit spécifiquement dominé par une classe d'échelle de conception. En effet les valeurs « mappées » peuvent être très diverses (hauteurs d'éléments, coordonnées de points, distances entre des objets, *etc*) comme les pertinences utilisées pour le mapping (réduire des hauteurs en fonction d'une échelle de voisinage, re-positionner des points en fonction d'une échelle parcellaire, re-dimensionner des éléments en bois en fonction d'une échelle économique ou technique, *etc.*).

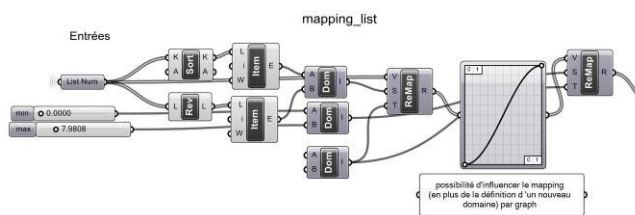
#### Evolution du sample et de son inscription dans les catalogues proposés



#### mapping\_list

→ transforme une liste de valeurs en une liste homologue mais comprise dans un domaine pouvant être redéfini

**GH v0.8.0006**  
**extrait\_valeur-min-max.ghx**



Inputs :  
Liste (A PLAT) de valeurs

Outputs :  
Liste mappée

Cf. Articles  
« Grasshopper et les arbres de données, introduction »  
« Le Pattern Mapping »

**Mapping**   **Arbres de données**

Exemples :

// Créer une surface  
dont la hauteur dépend  
de sa proximité à un  
point « répulseur », les  
distances entre la  
surface et le répulseur  
peuvent ensuite être  
« mappées » pour  
correspondre à des  
hauteurs min et max

GH v0.8.0006  
**créer\_srf\_repulseur.ghx**

**Figure 175 : mapping\_list (catalogue DNArchi-2012)**

Ce sample n'a été proposé que tardivement, il s'inscrit donc uniquement dans le catalogue DNArchi-2012. L'exemple créer\_srf\_repulseur est proposé : celui-ci montre comment mapping-list permet de contrôler les hauteurs maximales et minimales d'un modèle.

**Les mots clefs choisis pour caractériser le sample mapping-list sont :**

|  |                         |
|--|-------------------------|
| - relatifs à la culture :                            | <b>Mapping</b>          |
| - relatifs aux composants et fonctions utilisés :    | <b>Mapping</b>          |
| - relatifs aux notions sollicitées :                 | <b>Arbre de données</b> |
| - relatifs aux patterns impliqués :                  | <b>Mapping</b>          |
| - relatifs aux échelles dominant l'usage du sample : | <b>sans</b>             |

---

## Créer des points

### **3.4 Créer-trame-pts\_UV**

#### Spécificité du sample

Le code créer-trame-pts\_UV génère une trame orthogonale à partir d'une surface donnée (devant comprendre quatre arrêtes) et de paramètres U et V. De cette trame est obtenue les points par liste de quatre correspondants aux « sous-surfaces » créées par le découpage. Ce sample utilise le code extrait1item\_liste4 pour réorganiser la trame de points et la rendre spécialement utilisable pour instancier un composant (comme dans le code créer\_motif-maillage).

Ce code, comme le code Extrait\_1item\_liste4, peut être utilisé pour le pattern Place Holder pour créer un objet proxy à partir duquel diriger une partie de modèle (comme un motif). Mais créer-trame-pts\_UV peut également être utilisé pour mettre en œuvre le pattern Point Collection. Ce pattern (Woodbury 2010, p.212) vise à créer des points pour y instancier des éléments répétitifs. Ce pattern est

explicitement lié au pattern Place Holder mais s'en distingue toutefois. En effet le pattern Point Collection vise à créer des points, leur utilisation est au libre choix du concepteur. Et le pattern Place Holder n'utilise pas forcément des objets proxy « points » (il peut par exemple utiliser des polygones ou des segments), même si cela est possible.

Evolution du sample et de son inscription dans les catalogues proposés

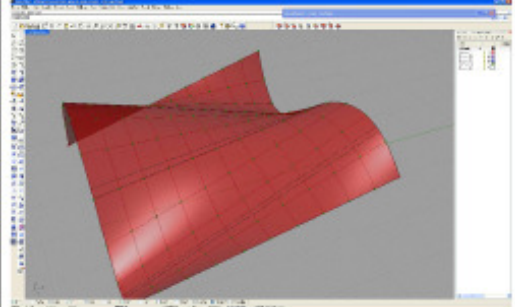
|   |  |  |  |
|---|--|--|--|
|  | <p><b>Creer_reseau-pts_UV</b></p> <p>Créer une collection de points sur une surface existante à partir de d'une division de la surface en U et V</p> | <p><i>Inputs :</i></p> <ul style="list-style-type: none"> <li>-Surface</li> <li>-Entier variable : V</li> <li>-Entier variable : U</li> </ul> <p><i>Output :</i></p> <ul style="list-style-type: none"> <li>- liste de points</li> <li>- liste d'arrêtes</li> <li>- liste de surfaces</li> </ul> | <p>Cours so823<br/>séance 2</p> <p>creer_reseau-pts_UV.ghx</p> |
|---|--|--|--|

Figure 176 : creer-trame-pts\_UV (catalogue So823-2011)

Ce code a été proposé dans le catalogue So823-2011 dans l'idée de permettre de créer rapidement un motif sur une surface. Il s'inscrit dans la séquence pédagogique « introduction à Grasshopper et aux arbres de données ».

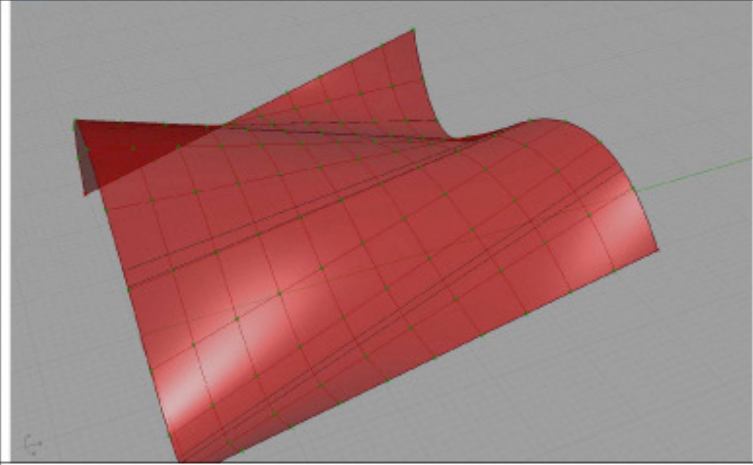
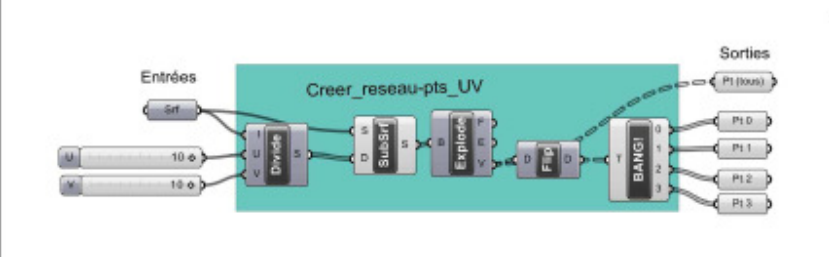
|   |   |  |
|---|---|--|
|  | <p><b>Créer_trame-pts_UV</b></p> <p>→ Crée une trame de point sur une surface (a 4 côtés) selon deux divisions (U et V)</p> <p>Inputs :<br/>Surface, Paramètres U et V</p> <p>Outputs :<br/>une trame de points</p> <p>Cf. SEANCE 2</p> | <p>GH v0.8.0006<br/><b>Creer_trame-pts_UV.ghx</b></p>  |
|  | <p><b>Exemples :</b></p> <p>// Créer une trame de pts sur une surface pour y « populer » un motif</p> <p>// Créer une trame de pts régulière et supprimer certains pts aléatoirement</p>  | <p>GH v0.8.0006<br/><b>creer_motif_maillage.ghx</b></p> <p>GH v0.8.0006<br/><b>Creer_pts-aleatoires_sur-trame-UV.ghx</b></p> |

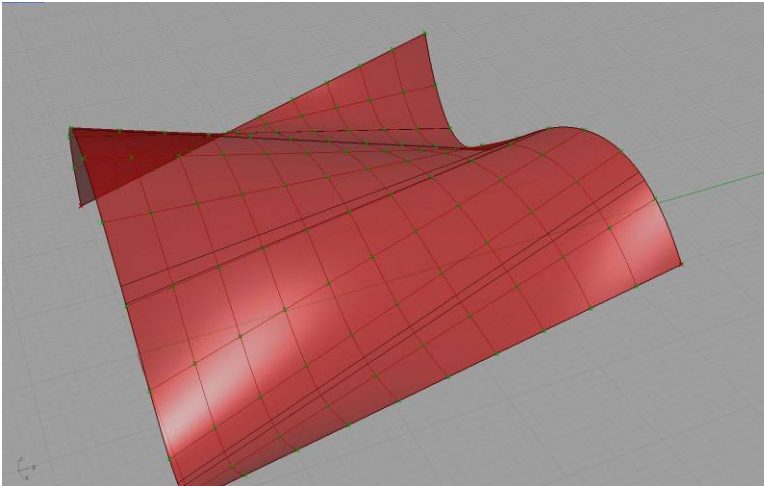
Figure 177 : creer-trame-pts\_UV (catalogue So823-2012)

Dans le catalogue So823-2012 des exemples viennent expliciter l'usage de ce sample:

-créer\_motif-maillage est relatif à la répétition d'un composant « maillage » sur la trame de points,

- créer\_pts-aleatoires\_sur-trame-UV crée une trame de points plus complexe, prenant en compte une part d'aléatoire

Sample et inscription retenue dans le catalogue DNArchi-2012



### Créer\_trame-pts\_UV

→ Crée une trame de point sur une surface (à 4 côtés) selon deux divisions (U et V)

Inputs :  
Surface,  
Paramètres U et V

Outputs :  
une trame de points

Cf. Articles  
« Le Pattern  
Place Holder »  
« Le Pattern  
Point  
Collection »

Exemples :  
// Créer une trame de pts sur une surface pour y « populer » un motif  
// Créer une trame de pts régulière et supprimer certains pts aléatoirement

GH v0.8.0006  
[Creer\\_trame-pts\\_UV.ghx](#)

GH v0.8.0006  
[creer\\_motif\\_maillage.ghx](#)

GH v0.8.0006  
[Creer\\_pts-aleatoires\\_sur-trame-UV.ghx](#)

[Place Holder](#)   [Point Collection](#)  
[Trame](#)   [Points « Sub Surface](#)

Figure 178 : creer-trame-pts\_UV (catalogue DNArchi-2012)



Dans le catalogue DNArchi-2012, le sample créer-trame-pts\_UV est proche de celui développé pour le catalogue So823-2012, mais permet cependant d'obtenir la trame de points sous différentes structures de données : -sous la forme de quatre listes (points 0, puis 1, puis 2, puis3), - sous la forme de listes correspondant aux colonnes, -sous la forme de listes correspondant aux lignes.

A la présentation du sample sont associées les notions des patterns Place Holder et Point Collection.

|   |                                       |
|---|---------------------------------------|
| <b>Les mots clefs choisis pour caractériser le sample créer-trame-pts_UV sont :</b> |                                       |
| - relatifs à la culture :   | <b>Motifs, Trame</b>                  |
| - relatifs aux composants et fonctions utilisés :                                   | <b>Points, « sub surface »</b>        |
| - relatifs aux notions sollicitées :  | <b>Place Holder</b>                   |
| - relatifs aux patterns impliqués :   | <b>Place Holder, Point Collection</b> |
| - relatifs aux échelles dominant l'usage du sample :                                | <b>sans</b>                           |

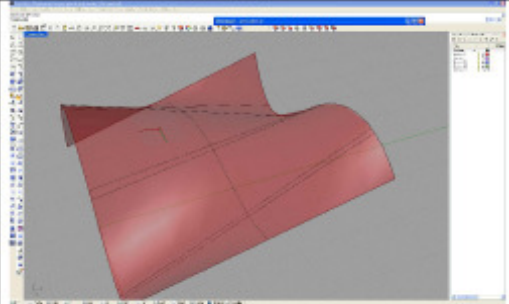
### 3.5 Creer\_pt\_UV

#### Spécificité du sample

Le code creer\_pt\_UV génère un point sur une surface en fonction de paramètres U et V. Dans sa version finale (DNArchi-2012), le sample s'appuie principalement sur le composant « evaluate point on surface » de Grasshopper.

Ce code a été développé pour la séquence « Attracteurs » du cours so823 de 2011 pour le code creer\_srf\_repulseur et réutilisé pour creer\_motif-reactif\_attracteur. Dans la mesure où ces deux codes font réagir des objets (une surface dans le premier cas, un motif dans le deuxième) à la proximité de points, ces codes correspondent explicitement au pattern reactor. Le pattern reactor (Woodbury 2010, p.230) « fait répondre un objet à la proximité d'un autre objet »<sup>118</sup>. Mais on peut imaginer que ce code puisse ici correspondre aussi au pattern controller, qui lui « contrôle (une partie de) modèle au travers un modèle simple indépendant »<sup>119</sup> (Woodbury 2010, p.191).

#### Evolution du sample et de son inscription dans les catalogues proposés

|   |  |   |  |
|---|--|---|--|
|  | <p><b>Creer_pt_UV</b></p> <p>Créer un point sur une surface existante en fonction de coordonnées en U et V</p> | <p><b>Inputs :</b></p> <ul style="list-style-type: none"> <li>-Surface</li> <li>-Entier variable : V (méthode 1)</li> <li>-Entier variable : U (méthode 1)</li> <li>-Slider 2D (méthode 2)</li> </ul> <p><b>Output :</b></p> <ul style="list-style-type: none"> <li>- un point</li> </ul> | <p>Cours so823<br/>séance 2</p> <p>creer-pt-uv.ghx</p> |
|---|--|---|--|

<sup>118</sup> « Make an object respond to the proximity of another object » (Woodbury 2010, p.230).

<sup>119</sup> « control (a part of) a model through a simple separate model. » (Woodbury 2010, p.191)



Figure 179 : créer\_pts\_UV (catalogue So823-2011)

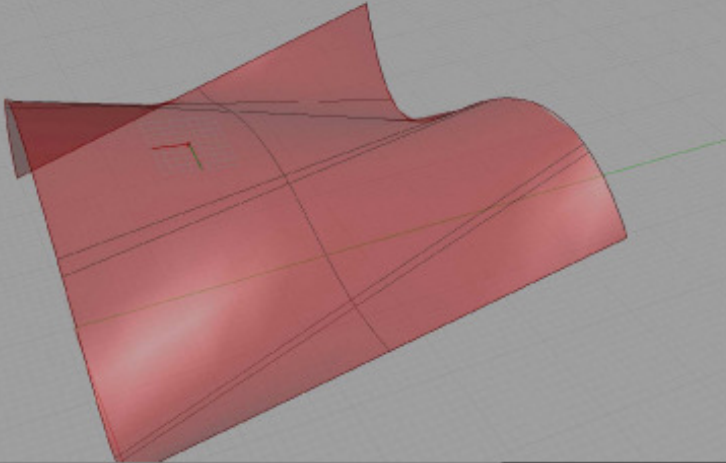
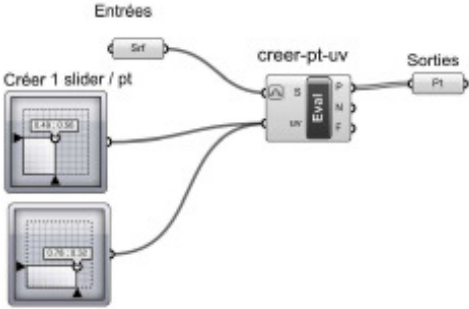
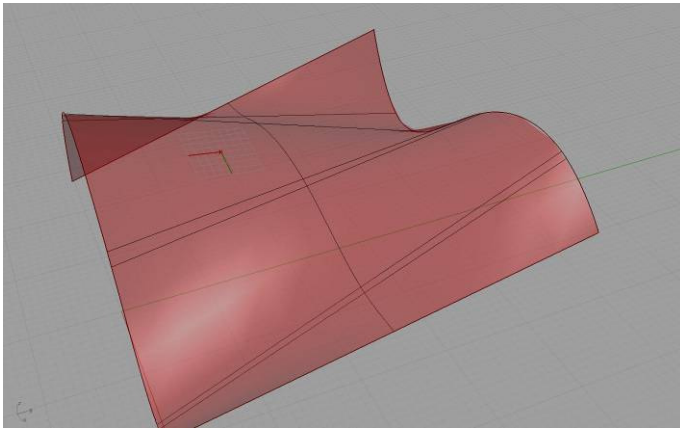
|  |   |   |
|--|---|---|
|   | <p><b>Creer_pt_UV</b></p> <p>→ Crée 1 pt sur une surface à partir de paramètres U et V</p> <p>Inputs :<br/>une surface, des paramètres U et V</p> <p>Outputs :<br/>Un ou plusieurs pts</p> <p><b>Cf. SEANCE 7</b></p> | <p><b>GH v0.8.0006</b><br/><b>creer-pt-uv.ghx</b></p>   |
|  | <p><b>Exemples :</b></p> <p>// Créer un (ou des) points aléatoires sur une surface donnée</p> <p>// Faire réagir un motif à un point défini sur la surface</p>  | <p><b>GH v0.8.0006</b><br/><b>Creer_pts-aleatoires_srf.ghx</b></p> <p><b>GH v0.8.0006</b><br/><b>Creer_motif-reactif_attracteur.ghx</b></p> |

Figure 180 : creer\_pts\_UV (catalogue So823-2012)

Les paramètres U et V définissant la position du point sur la surface sont contrôlés par des variables numériques dans le catalogue so823-2011 et par des variables graphiques dans le catalogue so823-2012. Le code est plus complexe dans sa version so823-2011 que dans la version so823-2012 où il correspond quasiment à l'utilisation d'un composant principal (le composant « evaluate point on surface »).

Dans le catalogue so823-2012 deux exemples intégrant le sample creer\_pt\_UV sont proposés : - creer\_motif-reactif\_attracteur et - créer\_pts-aleatoires\_srf.

Sample et inscription retenue dans le catalogue DNArchi-2012



## Creer\_pts\_UV

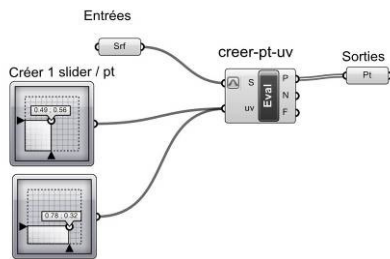
→ Crée n pt(s)  
sur une srf à  
partir de  
paramètres U et V

Inputs :  
Surface,  
Paramètres U et  
V

GH v0.8.0006  
[Creer\\_pts\\_UV.gh](#)  
x

Outputs :  
N pt(s)

Cf. Articles  
« Paramètres de  
surface et de  
points »  
« Le Pattern  
reactor »  
« Le Pattern  
Controller »



**reactor controller**

**Points UV « evaluate point on surface »**

Exemples :  
// Créer des  
points aléatoires  
sur une srf  
// faire réagir un  
motif à un point  
défini sur une  
surface

GH v0.8.0006  
[creer\\_pts-aleatoires\\_srf.gh](#)  
x

GH v0.8.0006  
[Creer\\_motif-reactif\\_attracteur.ghx](#)

Figure 181 : créer\_pts\_UV (catalogue DNArchi-2012)

Dans sa version pour DNArchi-2012, le sample creer\_pt\_UV est le même que celui développé pour le catalogue so823-2012.

**Les mots clefs choisis pour caractériser le sample creer\_pt\_UV sont :**

- relatifs à la culture : UV
- relatifs aux composants et fonctions utilisés : « evaluate point on surface » « point »
- relatifs aux notions sollicitées : sans
- relatifs aux patterns impliqués : Reactor, Controller

- relatifs aux échelles dominant l'usage du      sans  
sample :

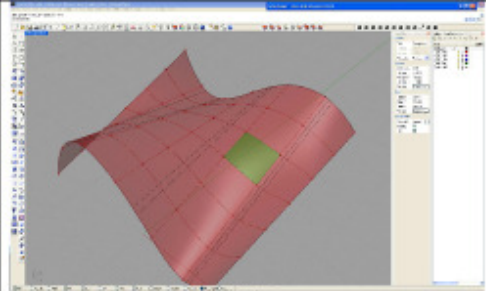
### 3.6 *creer\_liste-sous-srf\_UV*

Spécificité du sample

Le sample *creer\_liste-pts\_sous-srf-UV* permet de découper une surface de référence (construite sur Rhino comme sur Grasshopper) en sous-surfaces dont sont ensuite extraits les sommets, les arêtes et les faces. Ce sample s'appuie principalement sur les composants « SubSurface » et « explode » de Grasshopper.

Ce sample a été conçu en résonance avec un exercice récurrent dans les formations données par Bentley System pour Generative Components. Cet exercice découpe une surface en sous-surface dont sont extraites les arêtes sous forme de polygones. Sur ces polygones sont ensuite instanciés un composant. Le sample *creer\_liste-pts\_sous-srf-UV* permet de découper une partie de modèle pour lier facilement une autre partie de modèle comme un composant : ce sample est donc un exemple de code pouvant correspondre au pattern Place Holder mais pas au pattern Point Collection.

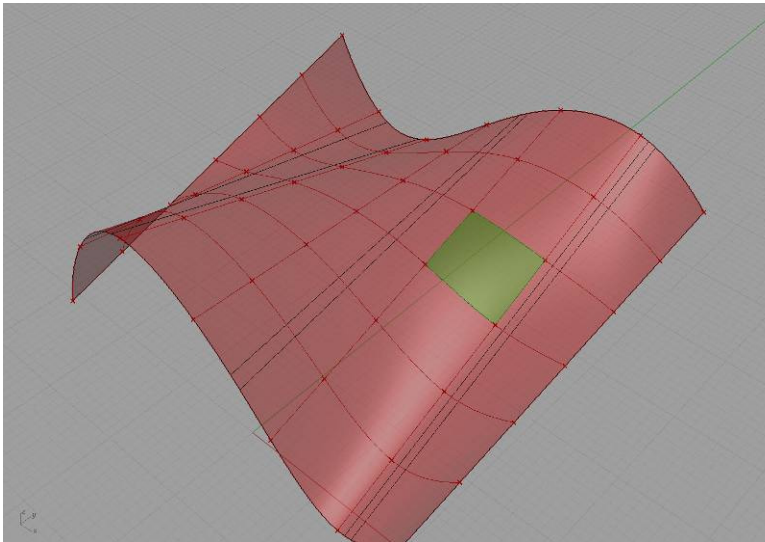
Evolution du sample et de son inscription dans les catalogues proposés

|   |  |   |   |
|---|--|---|---|
|  | <p><b>creer_liste-pts_sous-srf_UV</b></p> <p>Divise une surface en sous-domaines, à partir desquels on obtient des listes de points.</p> <p>(idem pattern <i>creer_reseau-pts_UV</i> )</p> | <p><i>Inputs :</i></p> <ul style="list-style-type: none"> <li>-Surface</li> <li>-Entier variable : V</li> <li>-Entier variable : U</li> </ul> <p><i>Output :</i></p> <ul style="list-style-type: none"> <li>- liste de points</li> <li>- liste d'arêtes</li> <li>- liste de surfaces</li> </ul> | <p>Cours so823<br/>séance 2</p> <p><i>creer_liste-pts_sous-srf_UV.ghx</i></p> |
|---|--|---|---|

**Figure 182 : *creer\_liste-sous-srf\_UV* (catalogue So823-2011)**

Le sample *creer\_liste\_sous-srf-UV* a été développé dans le catalogue so823-2011 en lien avec les codes *créer\_trame-pts-UV* et *créer\_motif-maillage*. Le code *creer\_liste-pts\_sous-srf-UV* permet d'extraire une plus grande diversité d'objets que *créer\_trame-pts-UV* : il permet d'obtenir les arêtes et les faces en plus des sommets. Cependant dans le cours so823 sont construits des exemples de « motifs » s'instanciant sur des points plutôt que sur des arêtes : le code n'a donc pas été intégré au catalogue so823-2012.

Sample et inscription retenue dans le catalogue DNArchi-2012



## Creer\_liste-sous-srf\_UV

→ Crée des sous surfaces à partir d'une surface d'origine et de paramètres U et V

GH v0.8.0006  
Creer\_pts\_UV.gh  
x

Inputs :  
Surface,  
Paramètres U et  
V

Outputs :  
N sous-surfaces  
et N arrêtes

Cf. Articles  
« Paramètres de  
surface et de  
points »  
« Le Pattern  
Place Holder »

Exemples :  
// « Populer » un  
composant sur  
une surface  
partir de son  
découpage en  
polygones

GH v0.8.0006

### Place Holder

Motifs Composants « Sub Surface »

Figure 183 : creer\_liste-sous-srf\_UV (catalogue DNArchi-2012)

Le sample creer\_liste\_sous-srf-UV est intégré dans le catalogue DNArchi-2012 dans la mesure où le sample creer\_trame-pts-UV dans cette version du catalogue ne lui est pas complètement redondant et propose d'extraire les sommets des sous-surfaces sous différentes forme de structure de données.

Ce sample est intégré à la séquence pédagogique « Motifs : Morphing Vs composant Vs tissage ».

Les mots clefs choisis pour caractériser le sample creer\_liste\_sous-srf-UV sont :

- relatifs à la culture : Motifs

|  |                    |
|--|--------------------|
| - relatifs aux composants et fonctions utilisés :    | « Sub Surface »    |
| - relatifs aux notions sollicitées :                 | Motifs, Composants |
| - relatifs aux patterns impliqués :                  | Place Holder       |
| - relatifs aux échelles dominant l'usage du sample : | sans               |

### 3.7 creer\_pts-aleatoires\_srf-UV

#### Spécificité du sample

Le code creer\_pts-aleatoires\_srf-UV génère des points aléatoires sur une surface à partir d'une trame de points de laquelle est supprimée aléatoirement un nombre défini d'items.

#### Sample et inscription retenue dans le catalogue DNArchi-2012

Ce code s'appuyant surtout sur le composant de Grasshopper « reduce », il a été uniquement conservé comme un exemple du sample creer-trame-pts\_UV dans le catalogue so823-2012 et non comme un sample à part entière.

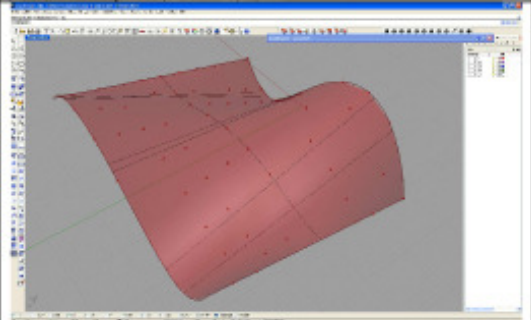
|  |  |  |  |
|--|--|--|--|
|  | <p><b>Creer_pts-aleatoires_sur-trame-UV</b></p> <p>Créer des points aléatoires sur une surface à partir d'une trame de points définis en U et V à laquelle on supprime aléatoirement des items</p> | <p><i>Inputs :</i></p> <ul style="list-style-type: none"> <li>-Surface</li> <li>-Entier variable : V</li> <li>-Entier variable : U</li> <li>- Nombre de points à supprimer</li> </ul> <p><i>Output :</i></p> <ul style="list-style-type: none"> <li>- liste de points</li> </ul> | <p>Cours so823<br/>séance 5</p> <p>Creer_pts-aleatoires_sur-trame-UV.ghx</p> |
|--|--|--|--|

Figure 184 : creer\_pts-aleatoires\_srf-UV (catalogue So823-2011)

### 3.8 creer\_pts\_aleatoires

#### Spécificité du sample

Le sample creer\_pts-aleatoires\_srf-UV permet de créer des points situés aléatoirement sur une surface donnée. Ce sample s'appuie sur le code creer\_pts\_UV mais génère des paramètres U et V aléatoirement grâce au composant « Random » de Grasshopper. Le nombre de points voulu est défini par une variable et l'aléatoire peut être recalculé. Le sample creer\_pts-aleatoires\_srf-UV a été développé à l'origine comme une partie du code creer\_crv-random-voronoi\_srf.

Etant donné ses possibles usages, ce sample peut être mis en relation avec le pattern Controller dont il peut être un fragment d'exemple.

#### Evolution du sample et de son inscription dans les catalogues proposés

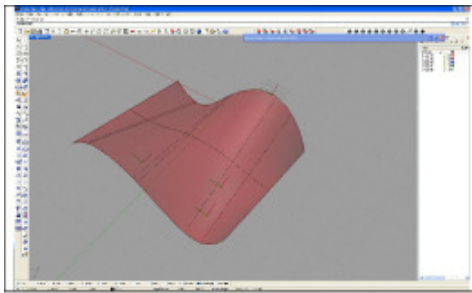
|   |   |  |                                 |
|---|---|--|---------------------------------|
|  | <p><b>Creer_pts-aleatoires</b></p> <p>Créer un nombre défini de points aléatoires sur une surface</p> <p>=&gt; utilise le pattern creer-pt-uv</p> | <p><i>Inputs :</i></p> <ul style="list-style-type: none"> <li>-Surface</li> <li>- Nombre de points à créer</li> </ul> <p><i>Output :</i></p> <ul style="list-style-type: none"> <li>- liste de points</li> </ul> | <p>Creer_pts-aleatoires.ghx</p> |
|---|---|--|---------------------------------|

Figure 185 : creer\_pts-aleatoires\_srf-UV (catalogue So823-2011)

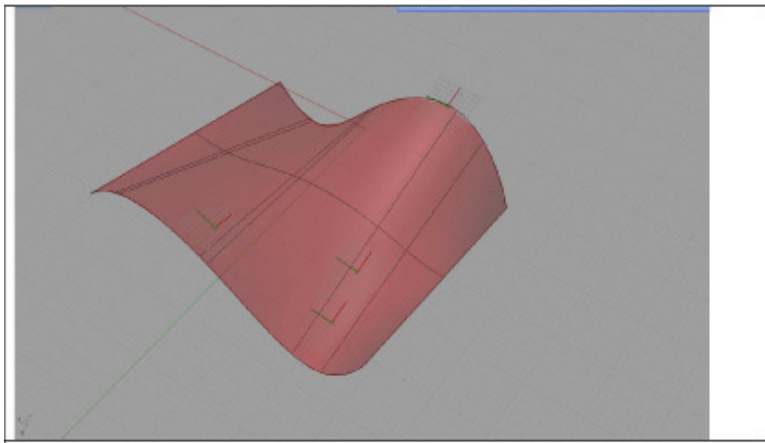
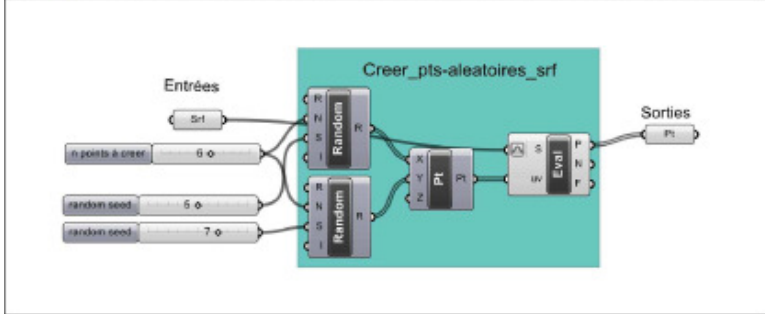
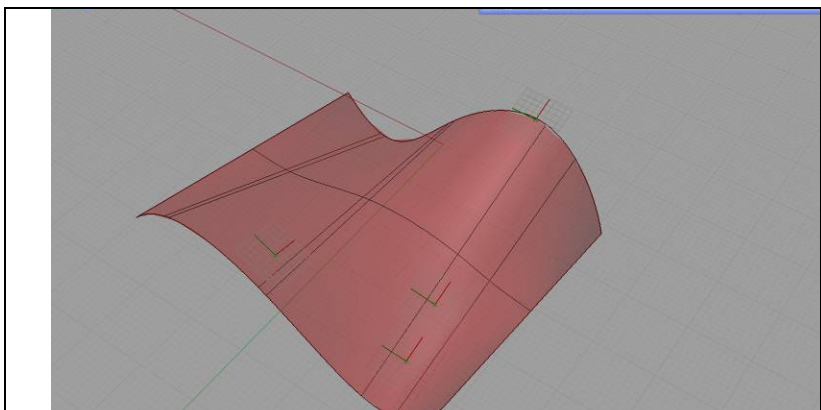
|  |   |   |
|--|---|---|
|   | <p><b>Creer_pts-aleatoires</b></p> <p>→ Crée des points aléatoires sur une surface donnée</p> <p><i>Inputs :</i><br/>Surface, nbr de pts voulus</p> <p><i>Outputs :</i><br/>Pts</p> <p>Cf. SEANCE 7</p> | <p>GH v0.8.0006<br/><b>Creer_pts-aleatoires.ghx</b></p>         |
|  | <p><b>Exemples :</b></p> <p>// Créer un diagramme de Voronoi sur une srf en fonction de n pts placés aléatoirement</p>  | <p>GH v0.8.0006<br/><b>creer_crv_random-voronoi_srf.ghx</b></p> |

Figure 186 : creer\_pts-aleatoires\_srf-UV (catalogue So823-2012)

Le code du sample creer\_pts-aleatoires\_srf-UV n'a pas changé entre les versions de 2011 et 2012. Néanmoins la version du catalogue so823-12 met le sample en relation avec un exemple d'application : le sample creer\_crv\_random-voronoi\_srf.

Sample et inscription retenue dans le catalogue DNArchi-2012

|   |  |   |
|---|--|---|
|  | <p><b>creer_pts-aleatoires_srf-UV</b></p> <p>→ Crée des points sur une surface à partir de paramètres U et V générés aléatoirement</p> | <p>GH v0.8.0006<br/><b>creer_pts-</b></p> |
|---|--|---|



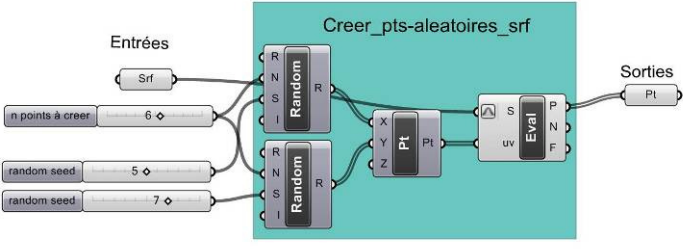
|  |   |   |
|--|---|---|
|  | <b>Inputs :</b><br>Surface, nbr de pts à générer<br><br><b>Outputs :</b><br>points                              | <b>aleatoires_srf-UV</b>                            |
|  | <b>Cf. Articles</b><br>« Paramètres de surface et de points »   |   |
|  | <b>Exemples :</b><br>// créer un diagramme de voronoi à partir de points aléatoirement répartis sur une surface | GH v0.8.0006<br><b>creer_crv-random-voronoi_srf</b> |
| <b>Random          Points</b>  |   |   |

Figure 187 : creer\_pts-aleatoires\_srf-UV (catalogue DNArchi-2012)

Le code creer\_pts-aleatoires\_srf-UV est le même que celui des versions so823-2011 et so823-2012. A ce sample est associé un article sur les paramètres des surfaces.

|  |                    |
|--|--------------------|
| <b>Les mots clefs choisis pour caractériser le sample creer_pts-aleatoires_srf-UV sont :</b> |                    |
| - relatifs à la culture :  | sans               |
| - relatifs aux composants et fonctions utilisés :  | « Random », Points |
| - relatifs aux notions sollicitées :   | sans               |
| - relatifs aux patterns impliqués :  | (controller)       |
| - relatifs aux échelles dominant l'usage du sample :   | sans               |

---

## Créer des courbes

### 3.9 Creer\_crv\_math\_spirale

#### Spécificité du sample

Le sample `creer_crv_math_spirale` permet de créer une courbe « spirale » à partir de son équation paramétrique. Ce sample s'appuie principalement sur le composant « fonction » et sur les équations paramétriques proposées par [mathcurve.com](http://mathcurve.com).

Ce sample peut être rapproché du `pattern point collection`, dont (Woodbury 2010, p.213) un des samples proposés (Woodbury 2010, p.213) est « Spiral », qui propose de générer une liste de point à partir d'une spirale<sup>120</sup> définie par son équation. Il est étonnant que nous ayons intuitivement affilié ce sample à la création de courbe, là où R. Woodbury y voit comme intérêt la création de points. Dans le catalogue DNachi-2012 nous conservons ce sample dans la catégorie « créer courbe » car ce code est pour nous lié à une séquence pédagogique d'apprentissage du scripting de courbes et de surface à partir d'équations paramétriques. Cependant dans les mots clefs permettant d'organiser les samples, il se trouve à la fois dans « points » et « courbes ».

Pour ce sample, les échelles dominantes semblent pouvoir être à la fois l'échelle de modèle et l'échelle géométrique. Nous mettons de côté l'échelle d'extension pour l'instant, car le code tel qu'il est proposé ne semble pas pouvoir directement être manipulé ainsi.

#### Evolution du sample et de son inscription dans les catalogues proposés

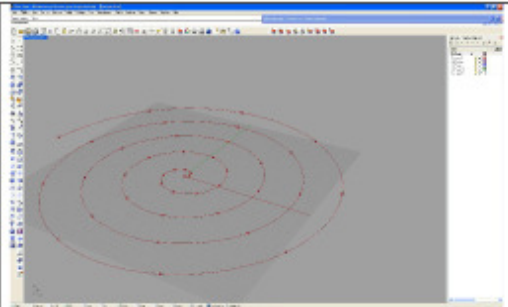
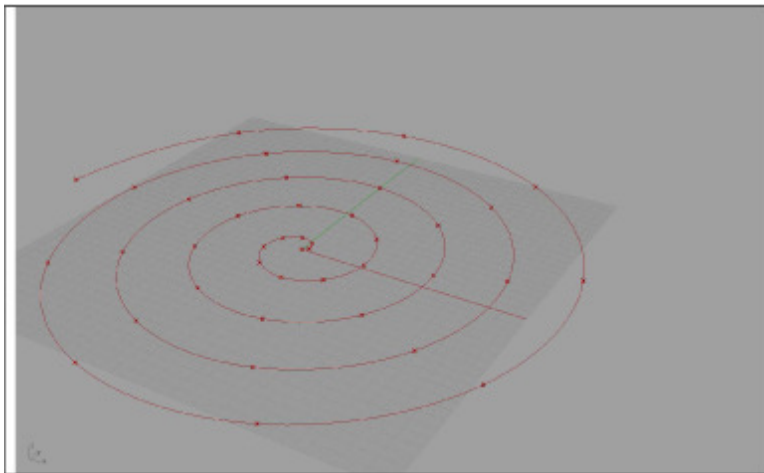
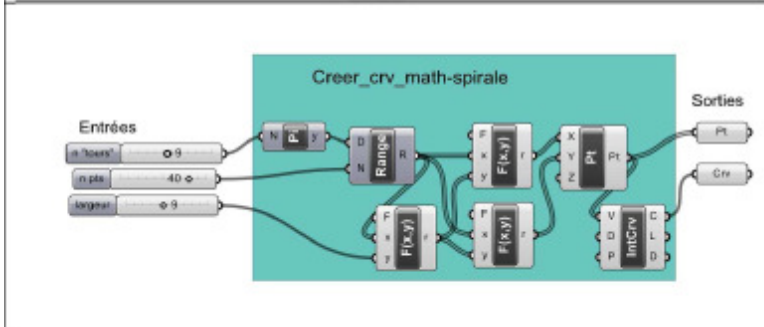
|   |   |  |  |
|---|---|--|--|
|  | <b>Creer_crv_math-spirale</b><br>Créer une spirale à partir de la formule mathématique de la courbe | <b>Inputs :</b><br>- Nombre de « tours »<br>- largeur<br>- nbr de points<br><br><b>Outputs :</b><br>-Points<br>-Courbe | Cours so823<br>séance 4<br><br><a href="http://www.mathcurve.com/">http://www.mathcurve.com/</a><br><br>Creer_crv_math-spirale.ghx |
|---|---|--|--|

Figure 188 : `creer_crv_math_spirale` (catalogue So823-2011)

---

<sup>120</sup> « Place a sequence of points along a spiral. » (Woodbury 2010, p.213).



|   |  |   |
|---|--|---|
|  | <p><b>Creer_crv_math-spirale</b></p> <p>→ Créer une spirale à partir de sa formule mathématique</p> <p>Inputs<br/>Nbr de « tours », largeur</p> <p>Outputs<br/>Pts et courbes</p> <p>Cf.<br/><a href="http://www.mathcurve.com">www.mathcurve.com</a><br/>Cf. SEANCE 8</p> | <p style="text-align: center;">GH v0.8.0006<br/><b>Creer_crv_math-spirale.ghx</b></p> |
|  | <p>//////////</p>  |   |

**Figure 189 : creer\_crv\_math\_spirale (catalogue So823-2012)**

Le catalogue so823-2012 n'apporte pas de changement majeur par rapport au catalogue so823-2011.

Ce sample, comme le sample suivant *creer\_srf-math-shell*, n'a pas été utilisé par les étudiants lors des expérimentations. Néanmoins il est proposé de conserver ces deux samples pour le catalogue DNArchi-2012 car ils illustrent le scripting de courbes grâce à leurs équations, ce qui reste un objectif pédagogique intéressant pour l'apprentissage de la programmation.

Sample et inscription retenue dans le catalogue DNArchi-2012

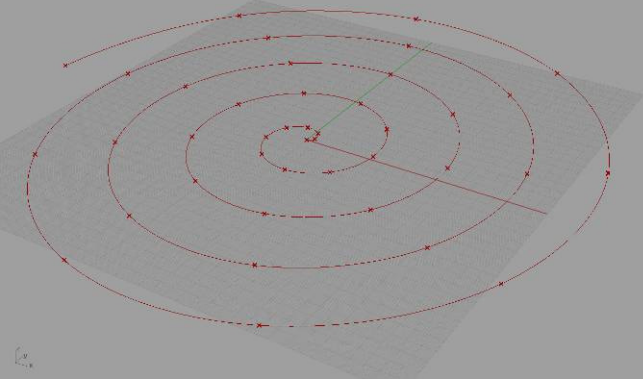
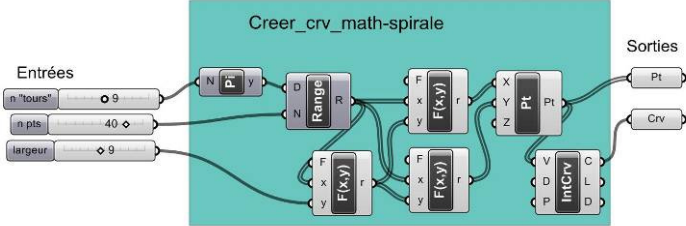
|   |  |  |
|---|--|--|
|    | <p><b>creer_crv_math_spirale</b><br/>→ crée une spirale à partir de son équation mathématique</p> <p><b>Inputs :</b><br/>Nbr de tours, Nbr de pts générés, largeur</p> <p><b>Outputs :</b><br/>Points, courbes</p> | <p style="text-align: right;">GH v0.8.0006<br/><b>creer_crv_math_spirale</b></p> |
|   | <p><b>Cf. Articles</b><br/>« Equations paramétriques »</p> <p><a href="http://www.mathcurve.com">www.mathcurve.com</a></p>   | <p>Exemples :<br/>////</p>   |
| <p style="text-align: center;"><b>Point Collection</b></p> <p style="text-align: center;">courbe      Points</p> <p style="text-align: center;">équations paramétrique    Spirale</p> |  |  |

Figure 190 : creer\_crv\_math\_spirale (catalogue So823-2012)

les mots clefs choisis pour caractériser le sample creer\_crv\_math\_spirale sont :

- relatifs à la culture : spirale
- relatifs aux composants et fonctions utilisés : « fonction », Courbes, Points
- relatifs aux notions sollicitées : Equations paramétrique
- relatifs aux patterns impliqués : Point Collections
- relatifs aux échelles dominant l'usage du sample : Echelle de modèle, échelle géométrique

### 3.10 Créer\_crv\_math-spirale3D

#### Spécificité du sample

Le sample `creer_crv_math_spirale3d` permet de créer une courbe « spirale » se déployant en hauteur à partir de son équation paramétrique. Ce sample s'appuie principalement sur le composant « fonction » et sur les équations paramétriques proposées par [mathcurve.com](http://www.mathcurve.com/).

#### Evolution du sample et de son inscription dans les catalogues proposés

Les versions du code `creer_crv_math_spirale3d` des catalogues `so823-2011`, `so823-2012` et `DNArchi-2012` sont très proches. Ce sample est lié à la séquence pédagogique du cours `so823` pour l'apprentissage de la programmation de courbes et de surface à partir de leurs équations paramétriques. Comme le sample `creer_crv_math_spirale`, ce sample est lié au pattern `Point Collection` (cf. *infra* 1.4.1).

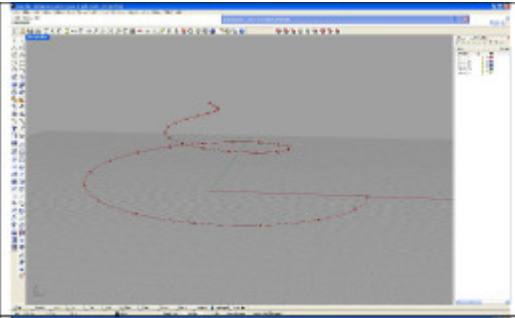
|  |  |  |   |
|--|--|--|---|
|  | <p><b>creer_crv_math-spirale3D</b></p> <p>Créer une spirale en 3 dimensions à partir de la formule mathématique de la courbe</p> | <p><b>Inputs :</b></p> <ul style="list-style-type: none"> <li>- Nombre de « tours »</li> <li>- largeur</li> <li>-nbr de points</li> <li>-hauteur max</li> </ul> <p><b>Outputs :</b></p> <ul style="list-style-type: none"> <li>-Points</li> <li>-Courbe</li> </ul> | <p>Cours so823<br/>séance 4</p> <p><a href="http://www.mathcurve.com/">http://www.mathcurve.com/</a></p> <p><code>creer_crv_math-spirale3D.ghx</code></p> |
|--|--|--|---|

Figure 191 : `creer_crv_math_spirale3d` (catalogue So823-2011)

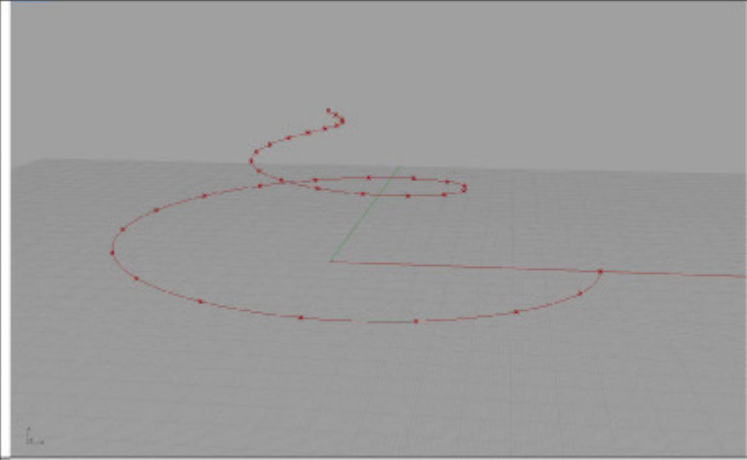
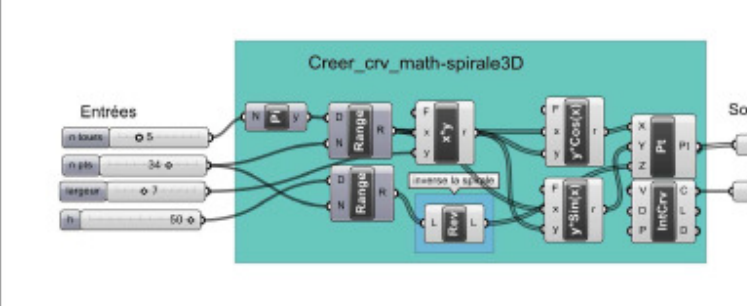
|   |   |   |
|---|---|---|
|  | <p><b>creer_crv_math-spirale3D</b></p> <p>→ Créer une spirale en 3 dimensions à partir de sa formule mathématique</p> <p><b>Inputs :</b><br/>Nbr de « tours », largeur, hauteur</p> <p><b>Outputs :</b><br/>Pts et courbes</p> <p>Cf.<br/><a href="http://www.mathcurve.com">www.mathcurve.com</a><br/>Cf. SEANCE 8</p> | <p>GH v0.8.0006<br/><b>Creer_crv_math-spirale3D.ghx</b></p> |
|  | <p>//////////</p>   |   |

Figure 192 : `creer_crv_math_spirale3d` (catalogue So823-2011)

Ce sample a été utilisé deux fois lors de l'expérimentation EXPso823-2011. Dans ces deux cas les appropriations ont néanmoins été sommaires (pas de transformation majeur du code, peu de modification des entrées).

Sample et inscription retenue dans le catalogue DNArchi-2012

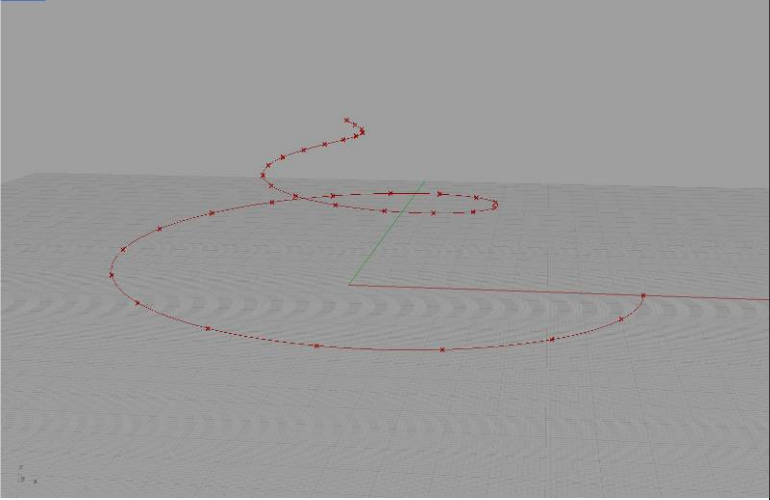
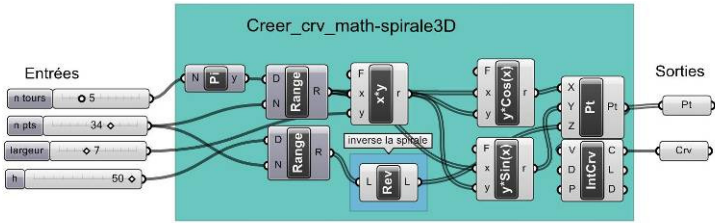
|   |  |   |
|---|--|---|
|                                  | <p><b>creer_crv_math-spirale3d</b><br/>→ crée une spirale en « 3 dimensions » à partir de son équation mathématique</p> <p>Inputs :<br/>Nbr de tours, Nbr de pts générés, largeur</p> <p>Outputs :<br/>Points, courbes</p> | <p>GH v0.8.0006<br/><b>creer_crv_math-spirale3d</b></p> |
|                                | <p><b>Cf. Articles</b><br/>« Equations paramétriques »<br/>« Paramètres de surface et de courbe »<br/><a href="http://www.mathcurve.com">www.mathcurve.com</a></p>   | <p><b>Exemples :</b></p>                                |
| <p><b>Point Collection</b>      <b>équations paramétrique</b></p> <p><b>courbe</b>              <b>Points</b></p> | <p><b>OCOPspiral.ghx</b></p> <p><a href="http://www.andrew.cmu.edu/org/tsunghsw-design/02_Patterns.html">http://www.andrew.cmu.edu/org/tsunghsw-design/02_Patterns.html</a></p>  |   |

Figure 193 : creer\_crv\_math-spirale3d (catalogue DNArchi-2012)

Dans le catalogue DNArchi-2012 le sample creer\_crv\_math-spirale3d est associé à deux articles : « Equations paramétriques » (cf. infra 1.4.1) et « Paramètres de surface et de courbe » afin de clarifier l'utilisation ici abusive du terme « spirale 3d ».

Dans les exemples, un lien vers le sample « Spiral » proposé par R. Woodbury est proposé.

Les mots clefs choisis pour caractériser le sample `creer_crv_math-spirale3d` sont :

|  |  |
|--|--|
| - relatifs à la culture :                            | spirale                                |
| - relatifs aux composants et fonctions utilisés :    | « fonction », courbe, points           |
| - relatifs aux notions sollicitées :                 | Equations paramétrique                 |
| - relatifs aux patterns impliqués :                  | Point Collection                       |
| - relatifs aux échelles dominant l'usage du sample : | Echelle de modèle, échelle géométrique |

### 3.11 *Creer\_crv\_math-balle-tennis*

#### Spécificité du sample

Le sample `creer_crv_math-balle-tennis` permet de générer une courbe correspondant au motif de la couture d'une balle de tennis, à partir de son équation paramétrique proposée par [mathcurve.com](http://mathcurve.com).

Le sample `creer_crv_math-balle-tennis` s'inscrit dans la même logique que les samples `creer_crv_math-spirale` et `creer_crv_math-spirale3d` (cf. *infra* 1.4.1). Il a particulièrement été développé en s'appuyant sur une anecdote rapportée par un architecte travaillant chez AJN. L'équipe d'AJN travaillant dans les premières phases de conception du musée Doha se serait appuyée sur une courbe semblable (échelle géométrique) pour simuler une rose des sables (échelle de modèle). Néanmoins ce travail n'avait pas été paramétrique faute de compétences disponibles à ce moment là. Dans la séquence pédagogique du cours `so823-2011`, la simulation de disque aux rayons aléatoires situé sur une courbe de « balle de tennis » était utilisée comme exemple.

Comme pour le sample `creer_crv_math-spirale`, `creer_crv_math-balle-tennis` est associé ici au pattern **Point Collection**.

#### Evolution du sample et de son inscription dans les catalogues proposés

Les versions `so823-2011` et `so823-2012` du sample `creer_crv_math-balle-tennis` sont les mêmes. Ce sample n'a pas été utilisé lors des expérimentations `so823-2011`, `so921-2011` et `so823-2012`.

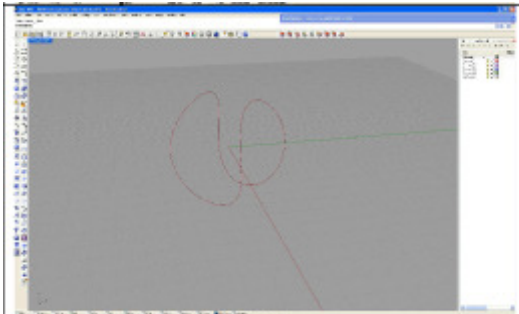
|   |  |   |   |
|---|--|---|---|
|  | <p><b>Creer_crv_math-balle-tennis</b></p> <p>Créer la courbe de la couture de la balle de tennis à partir de sa formule mathématique</p> | <p>Inputs :</p> <ul style="list-style-type: none"> <li>- Nombre de « tours »</li> <li>-nbr de points</li> <li>-variable a</li> <li>-variable b</li> </ul> <p>Outputs :</p> <ul style="list-style-type: none"> <li>-Points</li> <li>-Courbe</li> </ul> | <p>Cours <code>so823</code> séance 4</p> <p><a href="http://www.mathcurve.com/">http://www.mathcurve.com/</a></p> <p><code>creer_crv_math-balle-tennis.ghx</code></p> |
|---|--|---|---|

Figure 194 : creer\_crv\_math-balle-tennis (catalogue So823-2011)


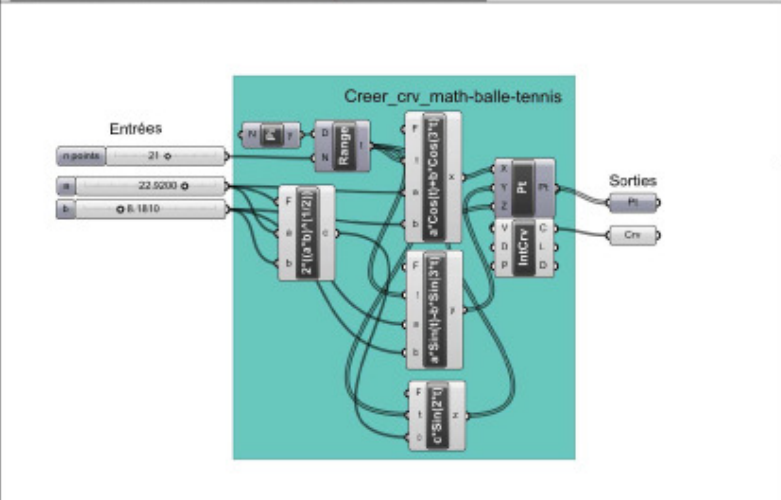
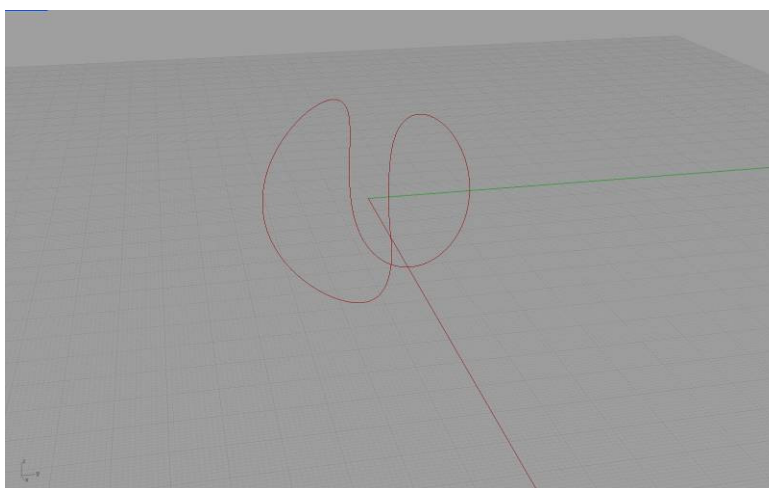
|  |  |  |  |
|--|--|--|--|
|   |  | <p><b>Creer_crv_math-balle-tennis</b></p> <p>→ Crée la courbe de la couture de la balle de tennis à partir de sa formule mathématique</p> <p>Inputs :<br/>Nbr de « tours », variables</p> <p>Outputs :<br/>Pts et courbes</p> <p>Cf.<br/><a href="http://www.mathcurve.com">www.mathcurve.com</a><br/>Cf. SEANCE 8</p> | <p style="text-align: center;">GH v0.8.0006<br/><b>creer_crv_math-balle-tennis.ghx</b></p> |
|  |  | <p>///////</p>   |  |

Figure 195 : creer\_crv\_math-balle-tennis (catalogue So823-2012)

Sample et inscription retenue dans le catalogue DNArchi-2012



**creer\_crv\_math-balle-tennis**

→ crée une courbe « couture de balle de tennis » à partir de son équation paramétrique

Inputs :  
Nbr de tours, variables

Outputs :  
Points, courbes

Cf. Articles  
« Equations paramétriques »

GH v0.8.0006  
**creer\_crv\_math-balle-tennis**



« La modélisation  
paramétrique  
chez AJN »  
[www.mathcurve.com](http://www.mathcurve.com)

Exemples :

**Point Collection**      **équations paramétrique**  
**courbe**                      **Points**

**Figure 196 : creer\_crv\_math-balle-tennis (catalogue DNArchi-2012)**

L'exemple du musée Doha (des disques aux dimensions aléatoires instanciés sur la courbe) est intégré dans le catalogue DNArchi-2012, il est associé à un article sur l'utilisation du paramétrique chez AJN.

|  |  |
|--|--|
| <b>Les mots clefs choisis pour caractériser le sample sont :</b> |  |
| - relatifs à la culture :  | sans                                   |
| - relatifs aux composants et fonctions utilisés :                | « fonction » points courbes            |
| - relatifs aux notions sollicitées :                             | Equations paramétriques                |
| - relatifs aux patterns impliqués :                              | Point Collection                       |
| - relatifs aux échelles dominant l'usage du sample :             | Echelle de modèle, échelle géométrique |

### **3.12 Creer\_crv\_sections-srf**

Spécificité du sample

Le code `creer_crv_sections-srf` permet de créer des courbes d'intersections entre une surface quelconque et des plans réguliers construits pour être verticaux et passer par des points de divisions réguliers des bords.

Ce code a été développé lors du cours So823-2011 pour l'exercice de synthèse « Waterloo ».

Evolution du sample et de son inscription dans les catalogues proposés



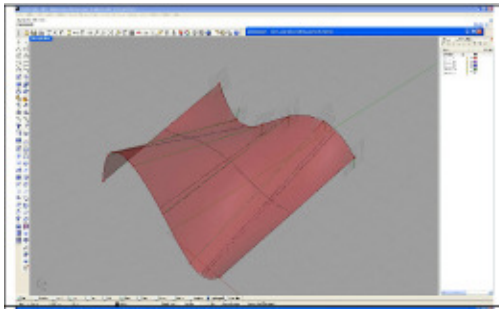
|   |   |   |  |
|---|---|---|--|
|  | <p><b>creer_courbes-sections_surfaces-sweep2</b></p> <p>Créer des courbes de « sections » d'une surface, en s'appuyant sur une division régulière de deux arêtes opposées de la surface</p> | <p><i>Inputs :</i></p> <ul style="list-style-type: none"> <li>-Surface</li> <li>- Courbe arrête (ou rail) 1</li> <li>- Courbe arrête (ou rail) 2</li> <li>-Entier variable : nombre de divisions</li> </ul> <p><i>Outputs</i></p> <ul style="list-style-type: none"> <li>-Courbes</li> <li>-Points</li> </ul> | <p>Cours so823 séance 9</p> <p>creer_crv-sections_surface-sweep2.ghx</p> |
|---|---|---|--|

Figure 197 : creer\_crv\_sections-srf (catalogue So823-2011)

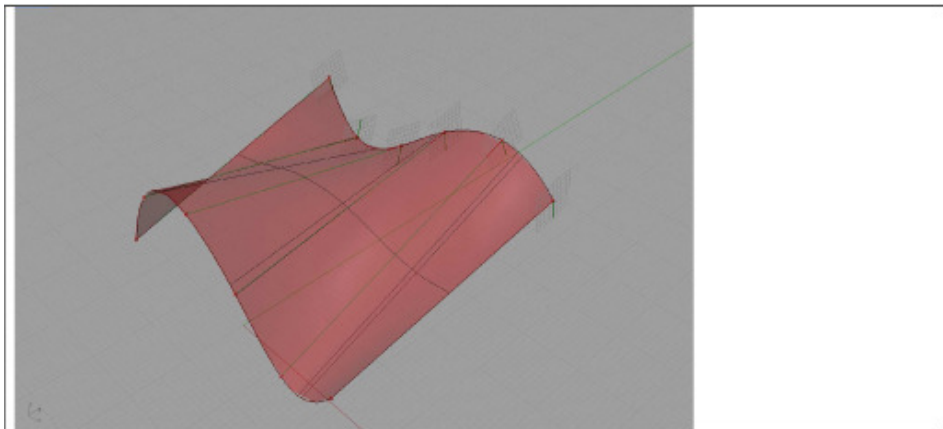
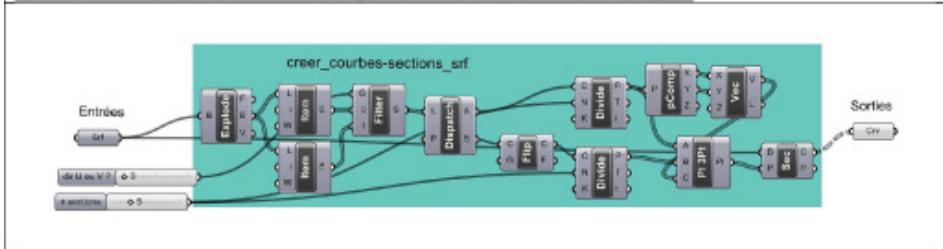
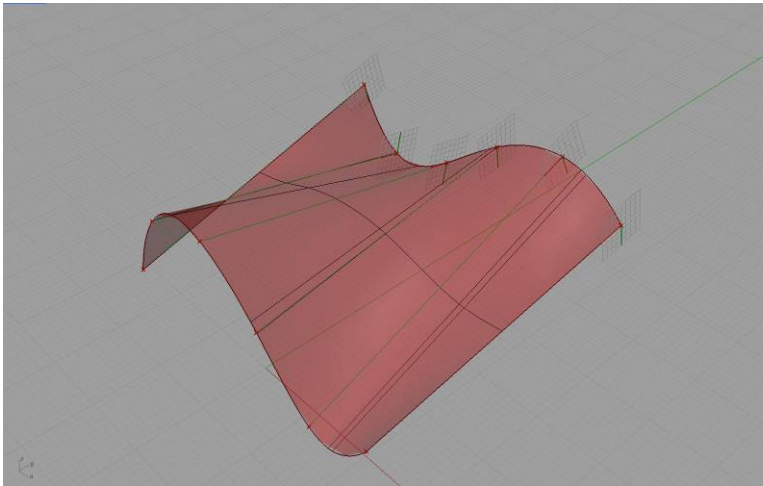
|  |  |  |
|--|--|--|
|   | <p><b>creer_courbes-sections_srf</b></p> <p>→ Créer des courbes de « sections » d'une surface</p> <p><i>Inputs :</i><br/>Surface, nbr de division, direction (U ou V) des sections</p> <p><i>Outputs :</i><br/>Courbes</p> | <p>GH v0.8.0006</p> <p><b>creer_courbes-sections_srf.ghx</b></p> |
|  | <p>Cf. SEANCE 9</p>  | <p>//////////</p>  |

Figure 198 : creer\_crv\_sections-srf (catalogue So823-2012)

Dans le catalogue so823-2012 le sample a été complexifié pour pouvoir permettre d'utiliser n'importe quelle surface et de contrôler l'orientation des sections (en U et V).

Sample et inscription retenue dans le catalogue DNArchi-2012





**creer\_crv\_sections-srf**  
 → crée des courbes de « sections » d'une surface

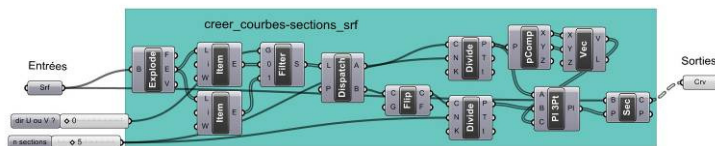
**Inputs :**  
 Srf, nbr de divisions, direction (U ou V) des sections

GH v0.8.0006  
 creer\_crv\_math-balle-tennis

**Outputs :**  
 courbes

**Cf. Articles**  
 « Tutoriel  
 « Waterloo » »

**Exemples :**



**courbe**      **Section Intersection**

**Figure 199 :** creer\_crv\_sections-srf (catalogue DNArchi-2012)

La version du sample DNArchi-2012 est la même que la version so823-2012. L'exemple du cours « Waterloo » a été transformé en tutoriel, ce tutoriel est lié à ce sample dans le catalogue DNArchi-2012.

|   |                              |
|---|------------------------------|
| <b>Les mots clefs choisis pour caractériser le sample creer_crv_sections-srf sont :</b> |                              |
| - relatifs à la culture :   | <b>Section</b>               |
| - relatifs aux composants et fonctions utilisés :                                       | <b>Intersection, courbes</b> |
| - relatifs aux notions sollicitées :  | sans                         |
| - relatifs aux patterns impliqués :   | sans                         |
| - relatifs aux échelles dominant l'usage du sample :                                    | sans                         |

### 3.13 Creer\_crv\_RandomVoronoi

Spécificité du sample

Le sample `creer_crv_random-voronoi` crée un diagramme de Voronoi sur une surface donnée à partir d'un certain nombre de points générés aléatoirement. Ce sample s'appuie beaucoup sur le composant « voronoi diagram » de Grasshopper, néanmoins il en pallie les limites puisque ce composant ne crée de diagramme de voronoi que sur un plan. Le sample projette ce diagramme sur la surface concernée.

Le diagramme de Voronoi générant un motif aujourd'hui iconique, ce sample semble être dominé par une échelle de modèle.

Evolution du sample et de son inscription dans les catalogues proposés

Les points qui permettent de définir le diagramme de Voronoi sont indiqués manuellement dans la version `so823-2011`. Dans la version `DNArchi-2012` les points sont générés aléatoirement dans l'exemple lié

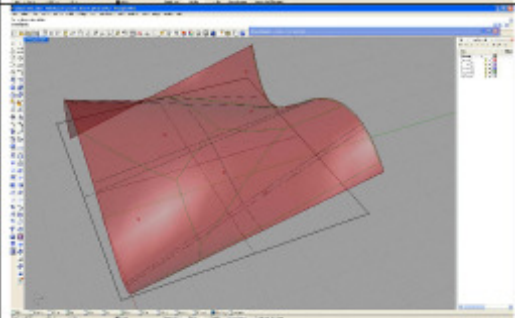
|   |  |   |   |
|---|--|---|---|
|  | <p><b>creer_crv_voronoi</b></p> <p>Créer un diagramme de voronoi sur une surface existante</p> | <p><i>Inputs :</i></p> <ul style="list-style-type: none"> <li>-Surface</li> <li>-Surface « emprise » sur le plan xy</li> <li>-Points sur la surface</li> </ul> <p><i>Outputs :</i></p> <ul style="list-style-type: none"> <li>-Points</li> <li>-Courbe</li> </ul> | <p>Cours so823<br/>séance 7</p> <p><code>creer_crv_voronoi.ghx</code></p> |
|---|--|---|---|

Figure 200 : `creer_crv_random-voronoi` (catalogue So823-2011)

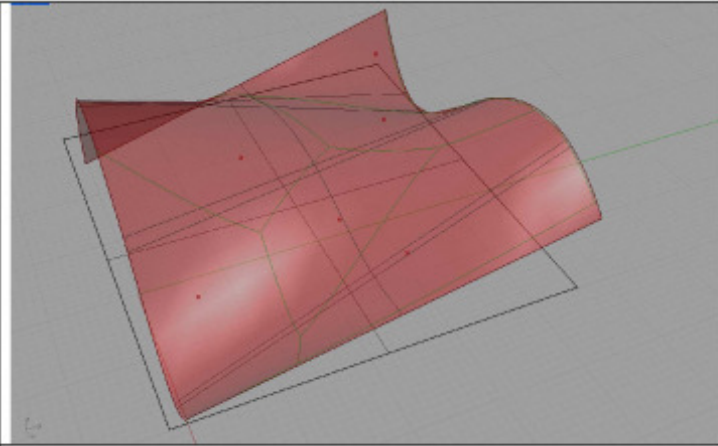
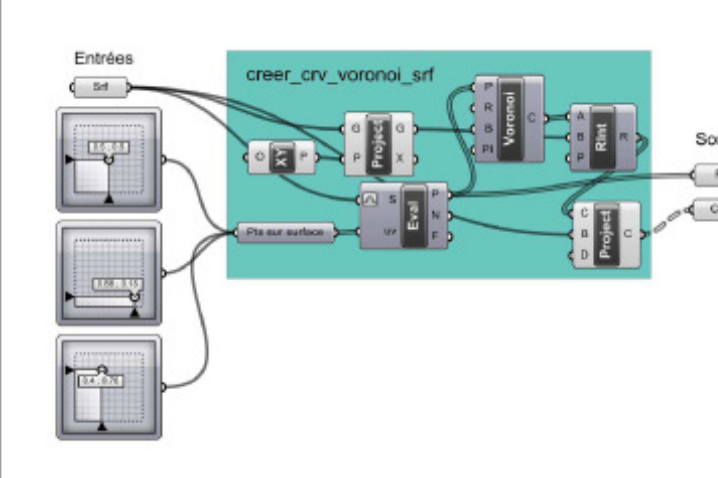
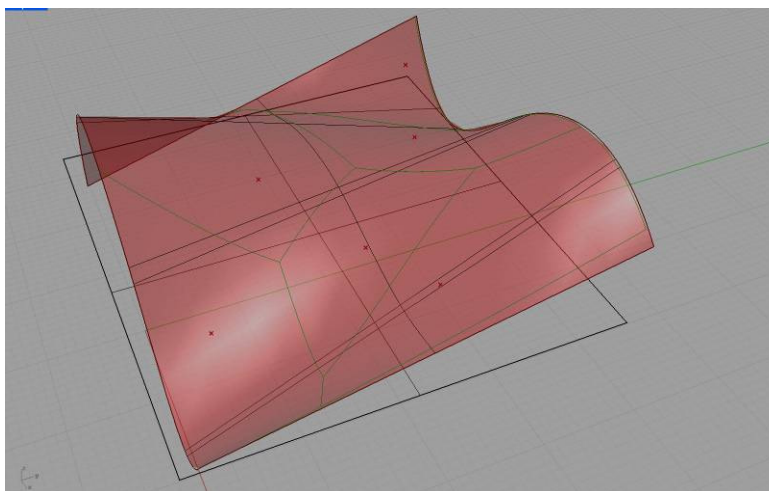
|   |  |   |
|---|--|---|
|  | <p><b>creer_crv-RandomVoronoi</b></p> <p>→ Crée un diagramme de voronoi sur une surface existante</p> <p><i>Inputs :</i></p> <p>Surface, nbr de pts sur la surface</p> <p><i>Outputs :</i></p> <p>Courbe</p> <p>Cf. SEANCE 6</p> | <p>GH v0.8.0006</p> <p><code>creer_crv-RandomVoronoi_srf.ghx</code></p> |
|  | <p>//////////</p>  |   |

Figure 201 : creer\_crv\_random-voronoi (catalogue So823-2012)

Sample et inscription retenue dans le catalogue DNArchi-2012



**creer\_crv\_rando  
m-voronoi**  
→ crée un  
diagramme de  
voronoi sur une  
srf existante

Inputs :  
Srf, nbr pts  
générant le  
diagramme

GH v0.8.0006  
**creer\_crv\_rando  
m-voronoi**

Outputs :  
courbes

Cf. Articles  
« V comme  
Voronoi »

Exemples :

**courbe**      **Voronoi**

**échelle(s) privilégiée(s) : échelle de modèle**

Figure 202 : creer\_crv\_random-voronoi (catalogue DNArchi-2012)

Dans le catalogue DNArchi-2012, ce sample est associé à un article d'Anne-Sophie Delaveau : « V comme Voronoi ». Cette association pointe des liens que l'on voudrait voir se multiplier entre la construction d'une culture de l'architecture numérique, culture construite selon des points de vue divers et des auteurs que l'on souhaite le plus nombreux possible, et des ressources visant à accompagner l'usage d'une technique.

**Les mots clefs choisis pour caractériser le sample sont :**

- |  |                                    |
|--|------------------------------------|
| - relatifs à la culture :                            | <b>Voronoi</b>                     |
| - relatifs aux composants et fonctions utilisés :    | <b>« voronoi diagram », courbe</b> |
| - relatifs aux notions sollicitées :                 | <b>Voronoi</b>                     |
| - relatifs aux patterns impliqués :                  | <b>sans</b>                        |
| - relatifs aux échelles dominant l'usage du sample : | <b>échelle de modèle</b>           |

## Créer des surfaces

### 3.14 *creer\_maillage\_random-delaunay*

#### Spécificité du sample

Le sample *creer\_maillage-random-delaunay* permet de créer, à partir d'une surface d'emprise, des points de hauteurs aléatoires comprises entre des hauteurs maximales et minimales définies par l'utilisateur. Un maillage de Delaunay est généré à partir de ces points.

Dans la mesure où ce sample génère des points ensuite utilisés pour créer le maillage, il peut être associé au pattern Point Collection.

L'utilisation d'une surface de référence marquant l'emprise du maillage généré, l'usage de ce sample semble dominé par l'échelle parcellaire.

#### Evolution du sample et de son inscription dans les catalogues proposés

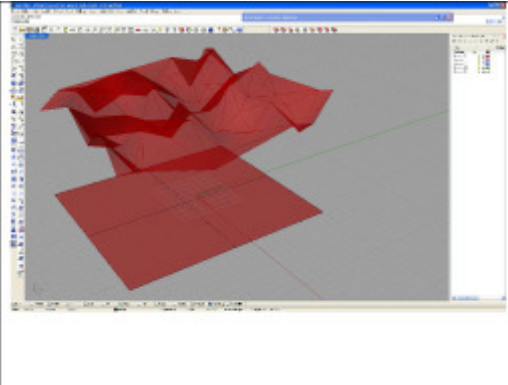
|  |  |  |  |
|--|--|--|--|
|  | <b>Creer_Surface-Projetee_RandomDelaunay</b><br><br>Créer une surface projetée à partir d'une emprise en plan, d'une hauteur maximum et d'une hauteur minimum, construite avec un diagramme de Delaunay généré aléatoirement | <b>Inputs :</b><br>-Surface<br>-Entier variable : V<br>-Entier variable : U<br>-Entier variable : nbre de points définissant le diagramme de Delaunay<br>-Entier variable : « graine » pour l'aléatoire<br>-Réal variable : Hmax<br>-Réal variable : Hmin<br><br><b>Outputs :</b><br>-Surfaces<br>-Points du diagramme | Cours so823<br>séance 7<br><br>Creer_Surface-Projetee_RandomDelaunay.ghx |
|--|--|--|--|

Figure 203 : *creer\_maillage\_random-delaunay* (catalogue So823-2011)

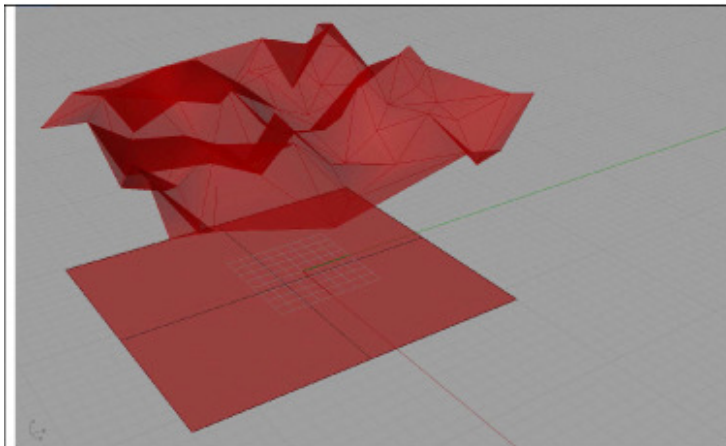
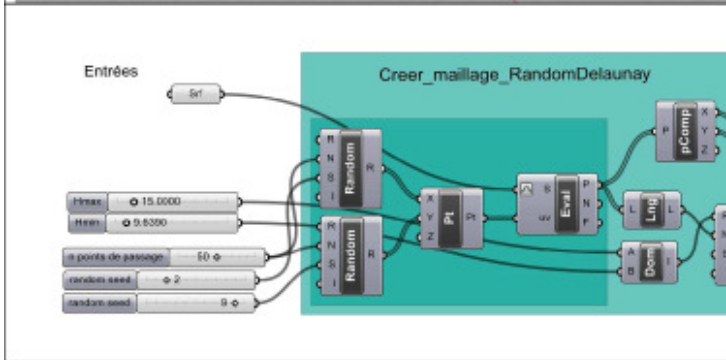
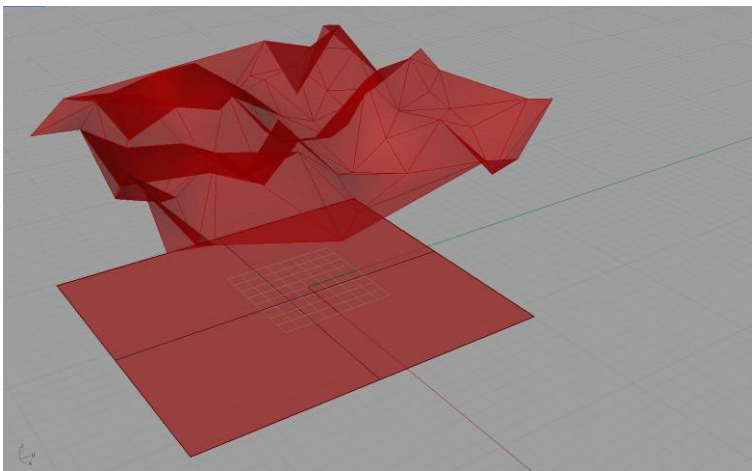
|  |  |  |  |
|--|--|--|--|
|   |  | <p><b>Creer_Maillage_RandomDelaunay</b></p> <p>→ Crée un maillage aléatoire à partir d'un diagramme de Delaunay</p> <p>Inputs :<br/>emprise au sol, hauteur maximum et hauteur minimum</p> <p>Outputs :<br/>Srf</p> <p><b>Cf. SEANCE 6</b></p> | <p>GH v0.8.0006<br/><b>Creer_Maillage_RandomDelaunay.ghx</b></p> |
|  |  |  |  |

Figure 204 : creer\_maillage\_random-delaunay (catalogue So823-2012)

Sample et inscription retenue dans le catalogue DNArchi-2012



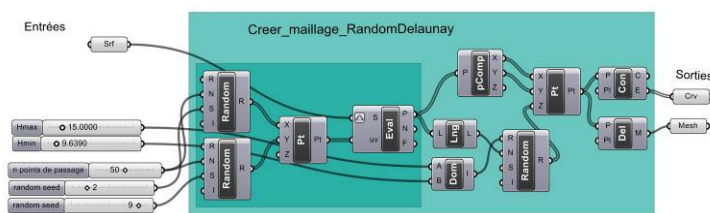
**creer\_maillage\_random-delaunay**

→ crée un diagramme de voronoi sur une srf existante

Inputs :  
Srf, nbr pts générant le diagramme

Outputs :  
courbes

GH v0.8.0006  
**creer\_crv\_rando-m-voronoi**



Cf. Articles  
« V comme  
Voronoi »

Exemples :

### Point Collection

Maillage Surface Delaunay Random

échelle(s) privilégiée(s) : échelle parcellaire

Figure 205 : creer\_maillage\_random-delaunay (catalogue So823-2011)

Les mots clefs choisis pour caractériser le sample créer\_maillage-random\_delaunay sont :

- relatifs à la culture : Delaunay, Random
- relatifs aux composants et fonctions utilisés : Mesh, Delaunay, Random
- relatifs aux notions sollicitées : sans
- relatifs aux patterns impliqués : **Point Collection**
- relatifs aux échelles dominant l'usage du sample : Echelle parcellaire

### 3.15 creer\_srf\_repulseur

Spécificité du sample

Le sample creer\_srf\_repulseur crée une surface dont la géométrie est liée à la proximité d'un point « répulseur ». Le mécanisme du sample vise à redéfinir les coordonnées en z d'une liste de points extraits d'une surface d'emprise (échelle parcellaire). Le calcul des nouvelles coordonnées en z est lié (op. de mise en relation) à la distance de ces points au point répulseur (cf. Reactor). Ces nouveaux points aux coordonnées en z modifiées (cf. Point Collection) sont utilisés pour générer la surface visée.

Ce sample est donc associé à une échelle parcellaire et à une opération de modélisation paramétrique de mise en relation. Il est également lié au pattern Reactor.



Ce sample est très proche d'un des samples proposé par R. Woodbury pour illustrer le pattern Reactor: le sample « Lift » qui « fait augmenter la longueur d'une ligne quand vous rapprochez un point de son origine » (Woodbury 2010, p.233). Ce sample correspond donc à un usage « classique » de la modélisation paramétrique. Il a été approprié à plusieurs reprises par les étudiants auxquels il a été proposé (cf. projet « topographies »)

Evolution du sample et de son inscription dans les catalogues proposés

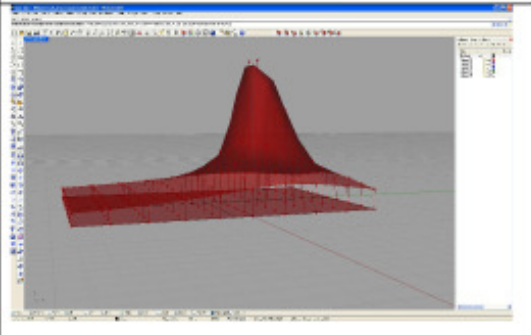
|   |   |  |  |
|---|---|--|--|
|  | <p><b>creer_surface-projetee_attracteur</b></p> <p>Créer une surface projetée à partir d'une empreise en plan, d'une hauteur et d'un point variable la faisant « réagir »</p> | <p><b>Inputs :</b></p> <ul style="list-style-type: none"> <li>-Surface</li> <li>-Entier variable : V=U</li> <li>- Réel variable : Coordonnées en x de l'attracteur</li> <li>- Réel variable : Coordonnées en y de l'attracteur</li> <li>- Réel variable : H max de la surface</li> </ul> <p><b>Outputs :</b></p> <ul style="list-style-type: none"> <li>-Surface</li> <li>-Points</li> </ul> | <p>Cours so823 séance 6</p> <p>creer_srf-projetee_attracteur.ghx</p> |
|---|---|--|--|

Figure 206 : creer\_srf\_repulseur (catalogue So823-2011)

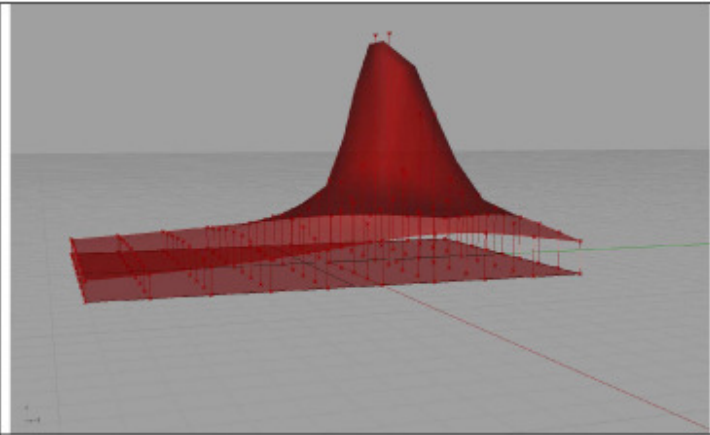
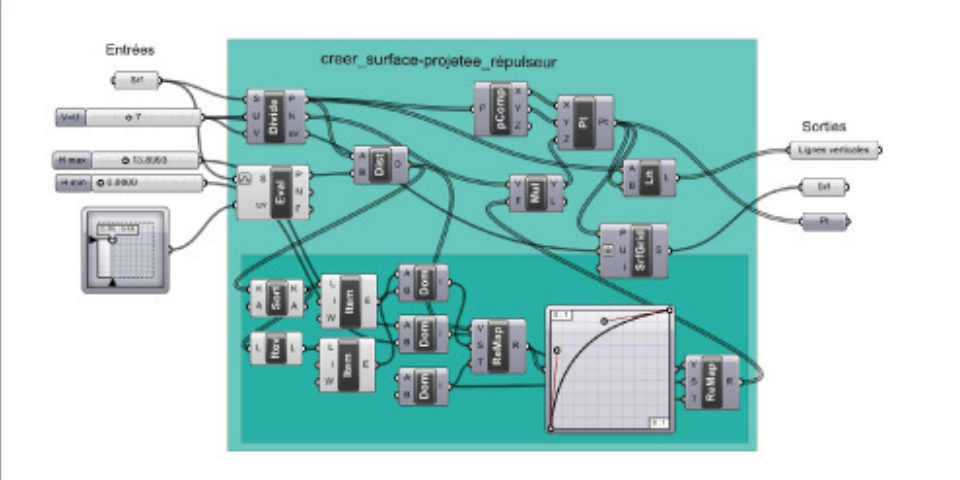
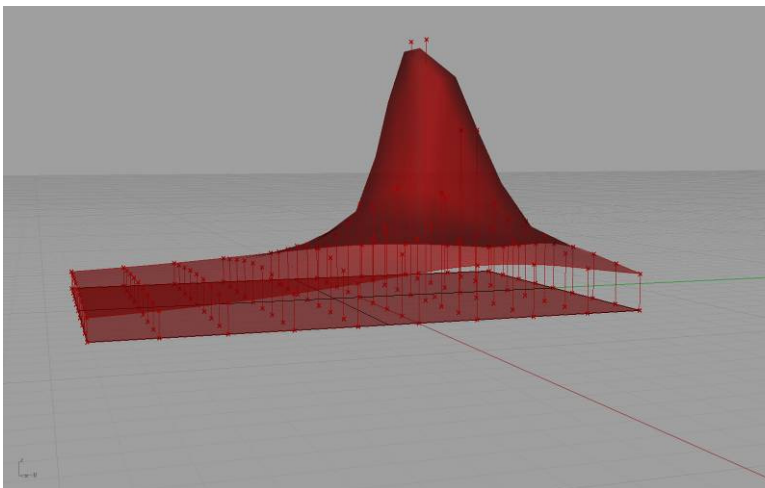
|  |   |   |
|--|---|---|
|    | <p><b>creer_surface-projetee_repulseur</b></p> <p>→ Crée une surface à partir d'une empreise en plan et de hauteurs variables en fonction d'un pt « repulseur »</p> <p>Inputs :<br/>empreise au sol, hauteurs max et min, UV du pt repulseur</p> <p>Outputs :<br/>Srf</p> <p>Cf. SEANCE 7</p> | <p>GH v0.8.0006<br/><b>creer_surface-projetee_repulseur.ghx</b></p> |
|  | <p><b>Exemples :</b></p> <p>// Créer une surface qui puisse réagir à plusieurs attracteurs</p> <p>GH v0.8.0006<br/><b>creer_surface-projetee_n-repulseurs.ghx</b></p>   |   |

Figure 207 : creer\_srf\_repulseur (catalogue So823-2012)

La version du code so823-2011 utilisait un point répulseur issu de la scène rhinocéros : celui-ci pouvait alors être placé hors de la surface. La version du code so823-2012 conserve le même principe de génération de points (Point Collection) en fonction d'une distance à un point de référence (cf. reactor). Mais il comprend néanmoins certaines modifications. Ainsi il utilise un point répulseur défini à partir du sample créer\_pt\_UV: le point répulseur est ainsi plus facilement définissable (il existe par défaut), néanmoins il ne peut être situé hors de la surface de référence. De plus la relation entre la distance au point répulseur et le coordonnée en z d'un point était contrôlé dans la version so823-2011 par un rapport r (distance \* r = z), cela rendait l'instanciation difficile : celle-ci ne pouvant être faite « qu'à l'œil », et ne permettant pas de vraiment contrôler les hauteurs maximales et minimales de la surface. Dans la version so823-2012 du pattern utilise le sample mapping-list (cf. infra. 1.2.3 et cf. Mapping) pour contrôler les valeurs minimales et maximales des coordonnées en z de la surface créée.

Sample et inscription retenue dans le catalogue DNArchi-2012



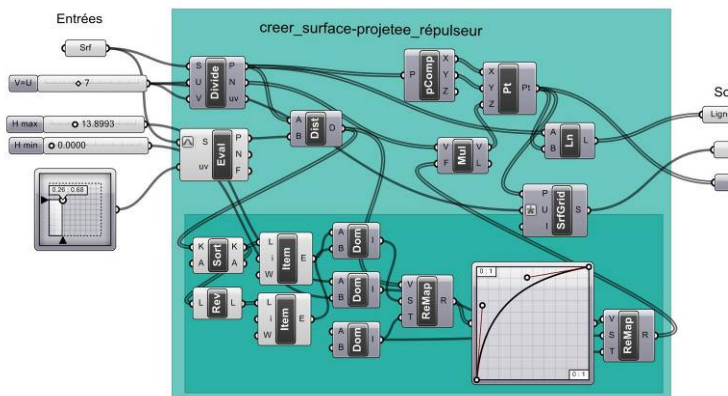
**créer\_srf\_repulseur**  
**ur** → crée une srf à partir d'une empreise au sol et de hauteurs réagissant à un pt « répulseur »

Inputs :  
 Srf d'empreise au sol, h max et min, UV

GH v0.8.0006  
**créer\_srf\_repulseur**  
**ur**

Outputs :  
 Srf, pts

Cf. Articles  
 « Pattern Reactor »  
 « Tutoriel surface réactive »  
 Exemples :





**Point Collection Reactor Mapping**

**Surface**

**échelle(s) privilégiée(s) : échelle parcellaire**

**Figure 208 : creer\_srf\_repulseur (catalogue DNArchi-2012)**

Pour la version DNArchi-2012 de ce sample (*cf.* fig. XXX), des références au tutoriel « surface reactive » et au pattern Rector sont intégrées. Les mots clefs proposés sont : Point Collection, Mapping et Rector (ce dernier est néanmoins plus important, en particulier c'est à l'article reactor que renvoient les notions liées), ainsi que le mot Surface. Une échelle associée de ce sample semble être l'échelle parcellaire.

Les mots clefs choisis pour caractériser le sample `creer_srf_repulseur` sont :

|  |   |
|--|---|
| - relatifs à la culture :                            | <b>reactor</b>                            |
| - relatifs aux composants et fonctions utilisés :    | <b>surface</b>                            |
| - relatifs aux notions sollicitées :                 | <b>sans</b>                               |
| - relatifs aux patterns impliqués :                  | <b>Reactor, Mapping, Point Collection</b> |
| - relatifs aux échelles dominant l'usage du sample : | <b>échelle parcellaire</b>                |

### **3.16 Creer\_srf\_sections**

Spécificité du sample

Le sample `creer_srf_sections` crée des intersections entre un volume et des plans horizontaux réguliers. Ce code a été créé pour permettre le calcul (*cf.* catégorie « évaluer ») de la superficie constructible d'un gabarit. Les versions so823-2011 et so823-2012 du sample demandent à l'utilisateur de créer une droite « hauteur maximale » pour construire les plans horizontaux. A partir de ce segment, deux méthodes sont proposées pour construire les plans symbolisant les niveaux : une méthode permettant d'utiliser une hauteur type d'étage, l'autre méthode permettant d'indiquer le nombre de niveaux s'insérant dans la hauteur maximale indiquée par le segment.

Evolution du sample et de son inscription dans les catalogues proposés

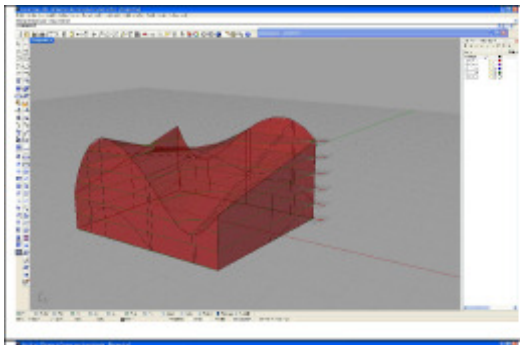
|   |  |  |   |
|---|--|--|---|
|  | <p><b>creer_srf_niveaux</b></p> <p>A partir d'un volume existant (BRep ou surface), créer des surfaces transversales à partir d'une verticale « maitre ». Ces niveaux peuvent être définis par deux méthodes : directement le nombre de niveau, ou la hauteur des niveaux.</p> | <p><b>Inputs :</b></p> <ul style="list-style-type: none"> <li>-Surface ou Brep</li> <li>-ligne « maitre » : verticale comprenant la hauteur maximale du volume</li> <li>-Entier variable : N sections</li> <li>-Réel variable : hauteur d'un niveau</li> </ul> <p><b>Outputs :</b></p> <ul style="list-style-type: none"> <li>- Surfaces</li> <li>- Courbes</li> </ul> | <p>Cours so823 séance 8</p> <p>creer_srf_niveau x.ghx</p> |
|---|--|--|---|

Figure 209 : creer\_srf\_sections (catalogue So823-2011)

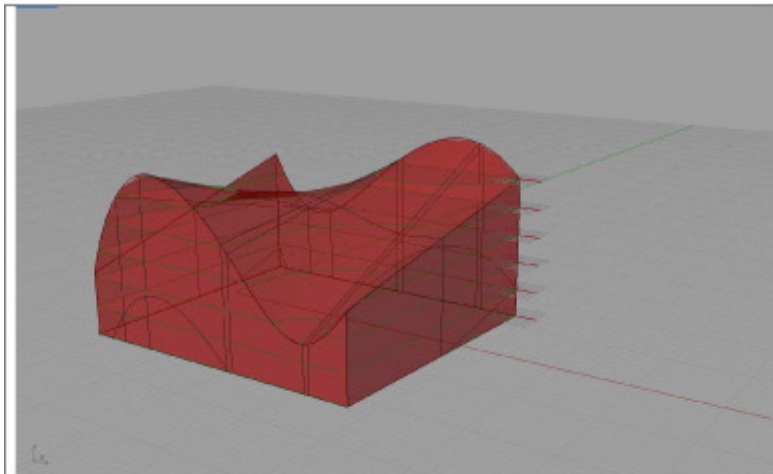
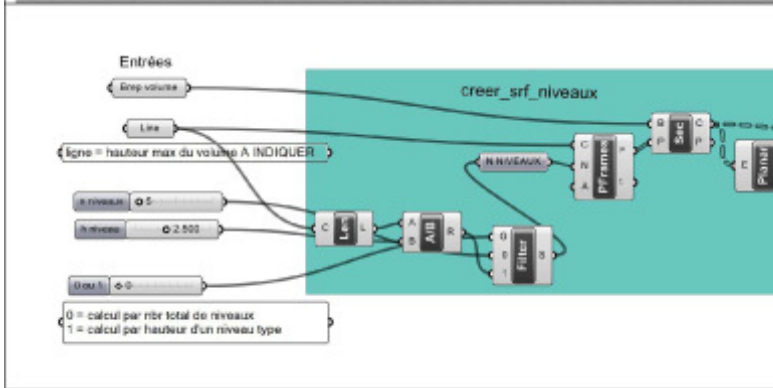
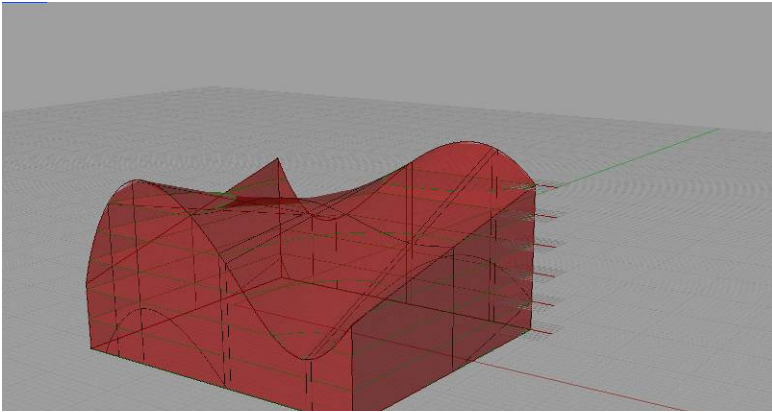
|   |  |   |  |
|---|--|---|--|
|   |  | <p><b>creer_srf-sections_volume</b></p> <p>→ Crée des niveaux sur un volume existant (2 méthodes : par nbr de niveaux ou par hauteur d'un niveau type)</p> <p><b>Inputs :</b><br/>Volume, nbr ou h des niveaux, « arrêtes principale »</p> <p><b>Outputs :</b><br/>Srf</p> <p><b>Cf. SEANCE 9</b></p> | <p>GH v0.8.0006<br/>creer_srf-sections_volu me.ghx</p> |
|  |  | <p><b>Exemples :</b><br/>// Calculer la srf constructible d'un gabarit</p>  |  |

Figure 210 : creer\_srf\_sections (catalogue So823-2012)

Sample et inscription retenue dans le catalogue DNArchi-2012



**creer\_srf\_sections**  
→ crée des niveaux sur un volume existant (2 méthodes possibles : par nbr de niveaux ou par hauteur type)

GH v0.8.0006  
creer\_srf\_section

**Inputs :** **S**  
Volume à évaluer,  
nbr ou h des  
niveaux

**Outputs :**  
srf

**Cf. Articles**

**Exemples :**

### **Evaluer surfaces constructibles      Intersections**

La version DNArchi-2012 de ce sample ne nécessite pas d'indiquer la hauteur maximale du volume au travers d'une droite. Le segment vertical à la hauteur maximale est directement calculé par le code. Ce code s'appuie sur le sample eval\_verticale-max.

**Les mots clefs choisis pour caractériser le sample creer\_srf\_sections sont :**

|   |   |
|---|---|
| - relatifs à la culture :                               | <b>Evaluer, surfaces constructibles</b> |
| - relatifs aux composants et fonctions utilisés :       | <b>Intersections</b>                    |
| - relatifs aux notions sollicitées :                    | <b>sans</b>                             |
| - relatifs aux patterns impliqués :                     | <b>sans</b>                             |
| - relatifs aux échelles dominant l'usage du<br>sample : | <b>sans</b>                             |

## **3.17 Creer\_srf\_mathShell**

### Spécificité du sample

Le sample creer\_srf\_math-shell s'inscrit dans le même contexte pédagogique que les samples creer\_crb-spirale *etc.* En effet ce sample appuie une séquence pédagogique qui vise à enseigner la modélisation de courbes et de surfaces à partir de leurs équations paramétriques.

L'échelle privilégiée de l'usage de ce sample semble être l'échelle de modèle.

Evolution du sample et de son inscription dans les catalogues proposés

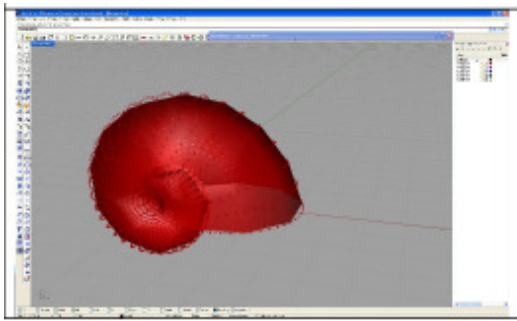
|   |   |   |   |
|---|---|---|---|
|  | <p><b>Créer_surfaces_math-shell</b></p> <p>Créer une surface « coquillage » à partir de sa formule mathématique</p> | <p><i>Inputs :</i><br/>-nbr de points<br/>-variable a</p> <p><i>Outputs :</i><br/>-Points<br/>-Courbe</p> | <p>Cours so823<br/>séance 5</p> <p>creer_srf_math-shell.ghx</p> |
|---|---|---|---|

Figure 211 : creer\_srf\_math-shell (catalogue So823-2011)

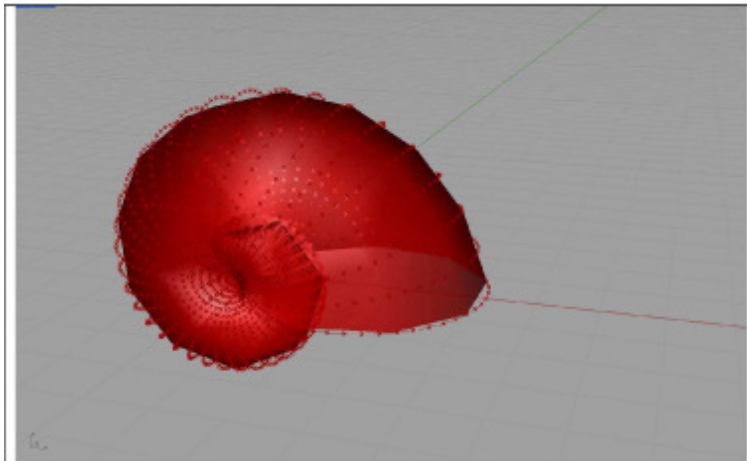
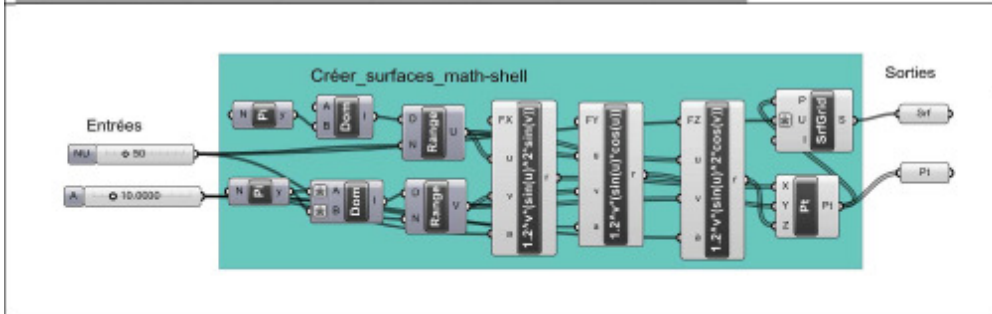
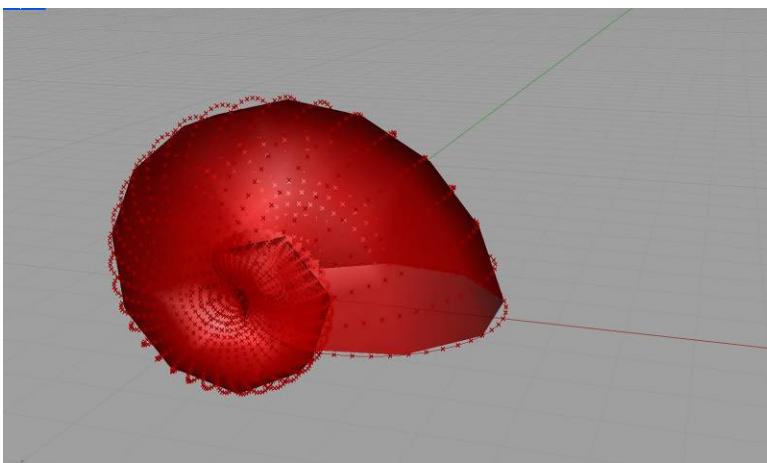
|  |  |  |   |
|--|--|--|---|
|    |  | <p><b>Créer_surfaces_math-shell</b></p> <p>→ Crée une surface « coquillage » à partir de sa formule mathématique</p> <p><i>Inputs :</i><br/>nbr de points et variable a</p> <p><i>Outputs :</i><br/>Srf</p> <p>Cf.<br/><a href="http://www.mathcurve.com">www.mathcurve.com</a><br/>Cf. SEANCE 8</p> | <p>GH v0.8.0006<br/><b>creer_srf_math-shell.ghx</b></p> |
|  |  | <p>Sorties</p> <p>Srf</p> <p>Pt</p>  |   |

Figure 212 : creer\_srf\_math-shell (catalogue So823-2011)

Les versions So823-2011, so823-2012 et DNArchi-2012 de ce sample sont très proches.

Sample et inscription retenue dans le catalogue DNArchi-2012



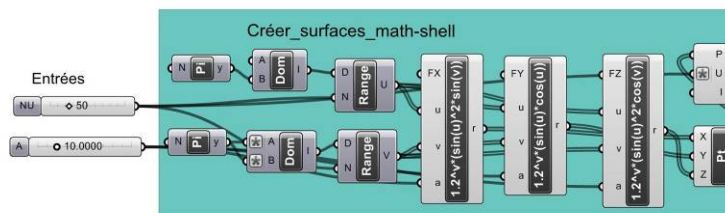
**creer\_srf\_math-shell**

→ crée une surface « coquillage » à partir de son équation paramétrique

*Inputs :*  
Nbr de pts,  
variable

GH v0.8.0006  
**creer\_srf\_math-shell**

Outputs :  
Points, surface



Cf. Articles  
« Equations  
paramétriques »  
[www.mathcurve.com](http://www.mathcurve.com)

Exemples :

Point Collection

Surface Points

équations paramétrique

échelle(s) privilégiée(s) : échelle de modèle ?

Figure 213 : creer\_srf\_math-shell (catalogue DNArchi-2012)

es mots clefs choisis pour caractériser le sample creer\_srf\_math-shell sont :

- |  |                        |
|--|------------------------|
| - relatifs à la culture :                            | Equations paramétrique |
| - relatifs aux composants et fonctions utilisés :    | « fonctions » points   |
| - relatifs aux notions sollicitées :                 | Equations paramétrique |
| - relatifs aux patterns impliqués :                  | Point Collection       |
| - relatifs aux échelles dominant l'usage du sample : | Echelle de modèle      |

## Créer des motifs

### 3.18 *creer\_motif-maillage*

#### Spécificité du sample

Le sample *creer\_motif-maillage* crée des cylindres formant un motif de treillis à partir d'une trame de points créée sur une surface existante. Ce sample fait partie d'une séquence pédagogique comprenant également les samples *creer\_trame-pts\_UV* et *extrait1item\_liste4*, utilisés dans la définition du motif de treillis.

Ce sample illustre tout à fait le pattern Place Holder de R. Woodbury. Il illustre également la séquence pédagogique « Motif : Morphing Vs composant Vs tissage ».

Ce sample étant assez explicitement lié à un motif de treillis et renvoyant donc à une structure, on peut le lier à une échelle technique.

#### Evolution du sample et de son inscription dans les catalogues proposés

Le samples *creer\_motif-maillage* version so823-2012 est différent de sa version so823-2011 : il reprend les modifications apportées au sample *extrait1item\_liste4* qui rend le code plus synthétique.

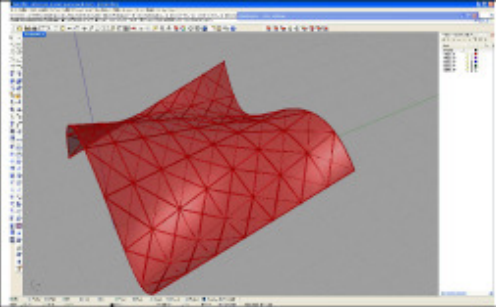
|   |   |   |   |
|---|---|---|---|
|  | <b>Creer_motif-maillage</b><br><br>Instancier un treillis sur une surface existante | <i>Inputs :</i><br>-Surface<br>-Entier variable : V<br>-Entier variable : U<br>-Réal variable : R<br><br><i>Output :</i><br>-Surfaces (maillage en croix à partir de « tube »)<br>-réseau de points | Cours so823<br>séance 2<br><br><i>creer_motif_mai<br/>llage.ghx</i> |
|---|---|---|---|

Figure 214 : *creer\_motif-maillage* (catalogue So823-2011)



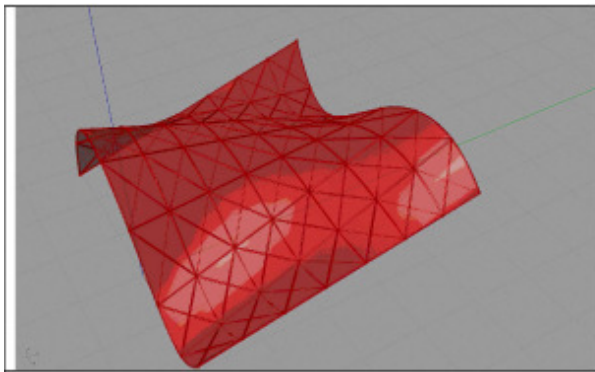
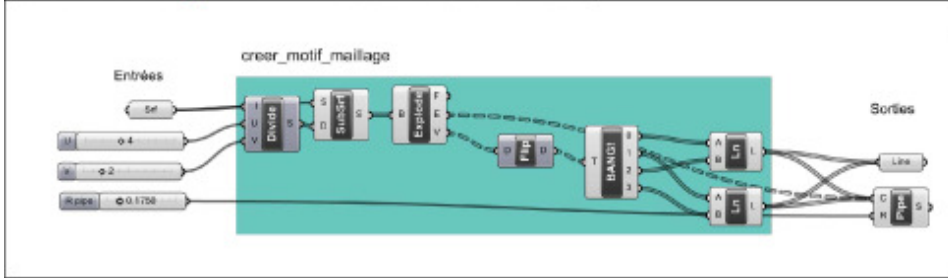
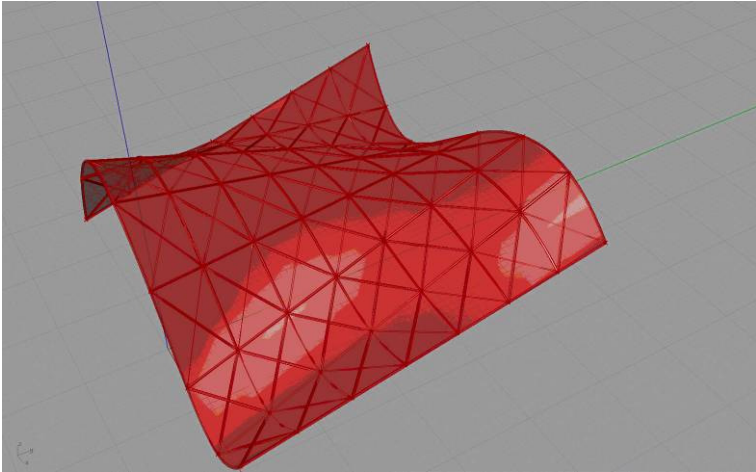
|  |  |  |   |
|--|--|--|---|
|   |  | <p><b>Créer_motif-maillage</b></p> <p>→ Crée un treillis sur une surface existante</p> <p>Inputs :<br/>Surface, nbr de divisions en U et V, R des cylindres</p> <p>Outputs :<br/>Srf</p> <p>Cf. SEANCE 2</p> | <p>GH v0.8.0006<br/><b>Créer_motif-maillage.ghx</b></p> |
|  |  |  |   |

Figure 215 : créer\_motif-maillage (catalogue So823-2012)

Sample et inscription retenue dans le catalogue DNArchi-2012

Le sample créer\_motif-maillage dans sa version DNArchi-2012 intègre un lien vers un tutoriel issu du cours so823-2013.



**créer\_motif-maillage**

→ crée un treillis sur une surface existante

Inputs :  
Srf, paramètres U et V définissant la trame, rayon des cylindres

GH v0.8.0006  
**créer\_motif-maillage**

Outputs :  
Courbes, surfaces

Cf. Articles



Exemples :

**Motif    Composant    Place Holder**

Figure 216 : creer\_motif-maillage (catalogue DNArchi-2012)

|   |                          |
|---|--------------------------|
| <b>Les mots clefs choisis pour caractériser le sample creer_motif-maillage sont :</b> |                          |
| <b>- relatifs à la culture :</b>  | <b>Motif , Composant</b> |
| <b>- relatifs aux composants et fonctions utilisés :</b>                              | <b>sans</b>              |
| <b>- relatifs aux notions sollicitées :</b>   | <b>sans</b>              |
| <b>- relatifs aux patterns impliqués :</b>  | <b>Place Holder</b>      |
| <b>- relatifs aux échelles dominant l'usage du sample :</b>                           | <b>Echelle technique</b> |

### **3.19 Creer\_motif-reactif\_attracteur**

#### Spécificité du sample

Le sample creer\_motif-reactif\_attracteur permet, à partir d'une surface de référence, de répéter des « motifs » sur une trame de points. Ces motifs sont modifiés en fonction de leur proximité à un point « attracteur » situé sur la surface. Ce code correspond à une séquence pédagogique du cours so823 : la séquence « Attracteur ». Ce sample se prête bien à diverses explorations : lors de cette séquence pédagogique, plusieurs exercices étaient proposés aux étudiants. Ces possibilités pourront être intégrées dans le catalogue DNArchi-2012 dans le tutoriel et/ou comme exemple du sample.

Ce sample correspond au pattern Reactor qui « *fait répondre un objet à la proximité d'un autre objet* » (Woodbury 2010, p.230). Ce sample est même très proche d'un sample proposé par R. Woodbury nommé « Circle Radii and Point Interactor » : ce code fait réagir le diamètre de cercles répétés sur une trame à la proximité à un point (Woodbury 2010, p.231).

#### Evolution du sample et de son inscription dans les catalogues proposés

Dans sa version so823-2011 le sample permet de répéter un motif hexagone dont la proximité au point « attracteur » modifie les dimensions.



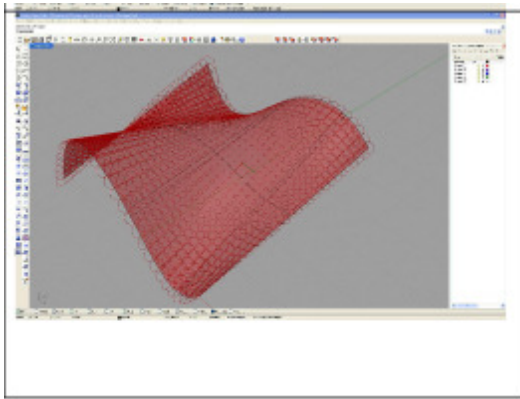
|   |   |  |   |
|---|---|--|---|
|  | <p><b>Créer_Motif-reactif_attracteurUV</b></p> <p>Répéter une figure sur une surface existante en la faisant réagir à un attracteur</p> | <p><i>Inputs :</i></p> <ul style="list-style-type: none"> <li>-Surface</li> <li>-Entier variable : V</li> <li>-Entier variable : U</li> <li>-Entier variable : nbre de points définissant le diagramme de Delaunay</li> <li>-Entier variable : « graine » pour l'aléatoire</li> <li>-Réel variable : Hmax</li> <li>-Réel variable : Hmin</li> </ul> <p><i>Outputs</i></p> <ul style="list-style-type: none"> <li>-Points</li> <li>-Motifs</li> </ul> | <p>Cours so823 séance 6</p> <p>creer_motif-reactif_attracteur.ghx</p> |
|---|---|--|---|

Figure 217 : creer\_motif-reactif\_attracteur (catalogue So823-2011)

Dans sa version so823-2012 le code mis à disposition à été complexifié : il permet d'utiliser comme motif n'importe quelle courbe plane et fermée, et permet également de définir des dimensions maximales et minimales du motif déformé (ce code utilise le sample mapping\_list).

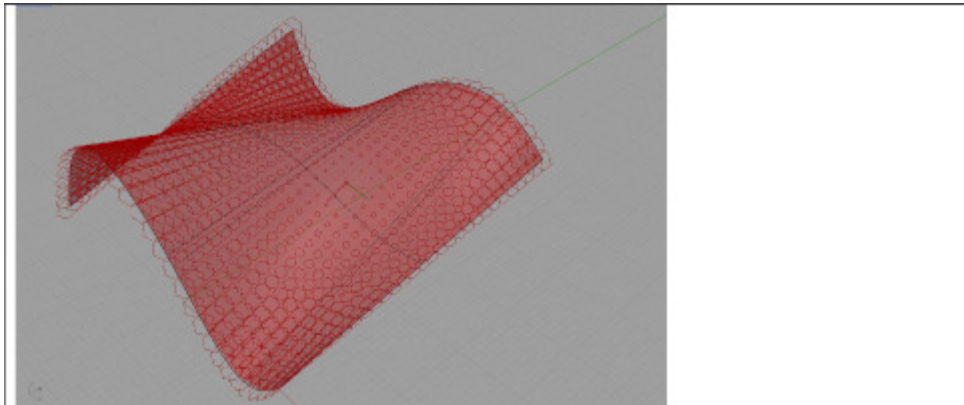
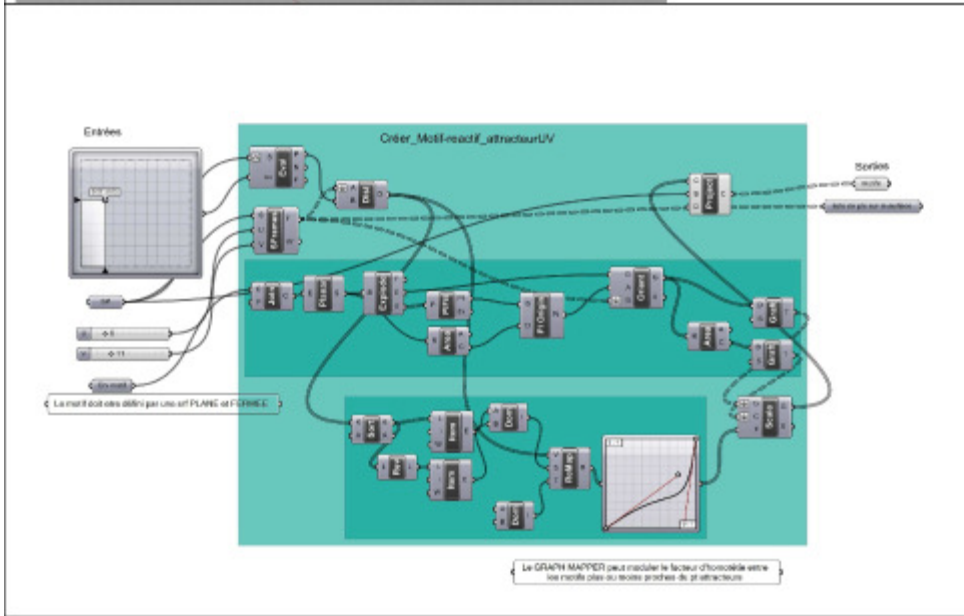
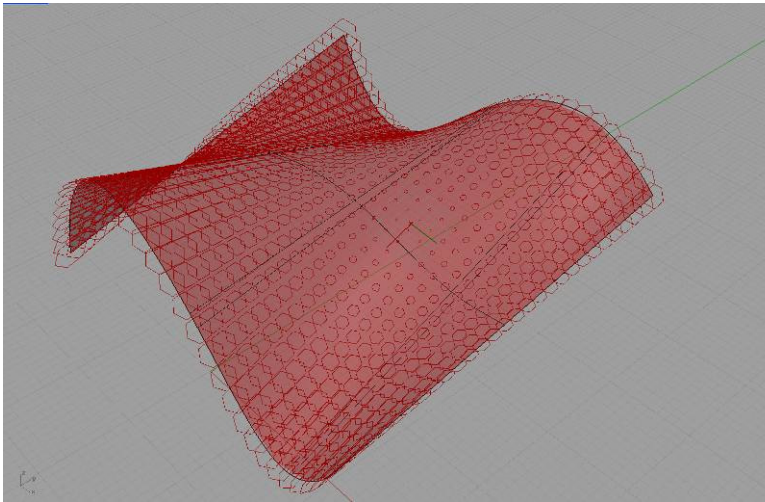
|  |  |  |
|--|--|--|
|   | <p><b>Créer_Motif-reactif_attracteurUV</b></p> <p>→ Répète un motif sur une surface donnée en fonction d'un pt « attracteur »</p> <p><i>Inputs :</i><br/>Srf, paramètres U et V d'un point, motif défini par une COURBE PLANE et FERMEE</p> <p><b>Cf. SEANCE 7</b></p> | <p>GH v0.8.0006<br/><b>creer_motif-reactif_attracteur.ghx</b></p>    |
|  | <p><b>Exemples :</b><br/>// Faire réagir un motif à n « attracteurs »</p>  | <p>GH v0.8.0006<br/><b>creer_motif-reactif_n-attracteurs.ghx</b></p> |

Figure 218 : creer\_motif-reactif\_attracteur (catalogue So823-2012)

Sample et inscription retenue dans le catalogue DNArchi-2012



**creer\_motif-reactif\_attracteur**  
 → répéter un motif sur une srf donnée réagissant à un pt « attracteur »

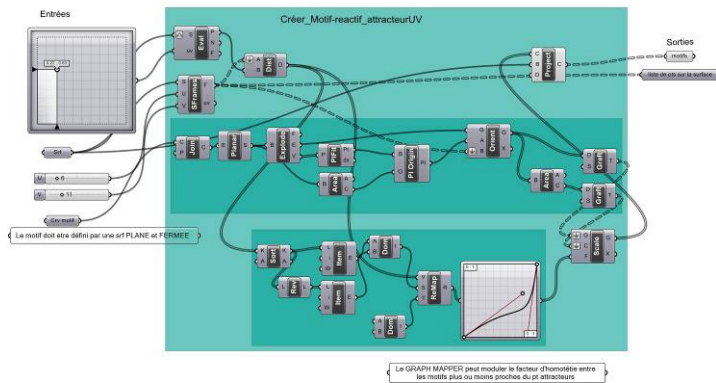
**Inputs :**  
 Srf, paramètres U et V définissant la trame, position du pt attracteur via un slider 2D

GH v0.8.0006  
**creer\_motif-reactif\_attracteur**

**Outputs :**  
 Courbes

*Cf. Articles*  
 « Tutoriel  
 « attracteur » »

**Exemples :**



**Motif attracteur Composant**

**Reactor Place Holder Mapping**

Figure 219 : creer\_motif-reactif\_attracteur (catalogue DNArchi-2012)

**Les mots clefs choisis pour caractériser le sample creer\_motif-reactif\_attracteur sont :**

- relatifs à la culture : **Motifs, reactor**
- relatifs aux composants et fonctions utilisés : **Re Map**
- relatifs aux notions sollicitées : **Mapping, composant, attracteur**
- relatifs aux patterns impliqués : **Reactor, Place Holder**
- relatifs aux échelles dominant l'usage du sample : **sans**

### 3.20 Creer\_motif-morphing

Spécificité du sample

Le sample creer\_motif-morphing permet de répéter un volume (le motif) sur une trame définie sur une surface de référence. Ce sample s'appuie particulièrement sur le composant « morphing » proposé par Grasshopper. Ce composant permet de répéter un volume, dont on définit une « boîte englobante », dans des « volumes cibles » définis sur la surface de référence en fonction d'une trame U et V et d'une hauteur. Ce sample correspond au Pattern Place Holder.

Evolution du sample et de son inscription dans les catalogues proposés

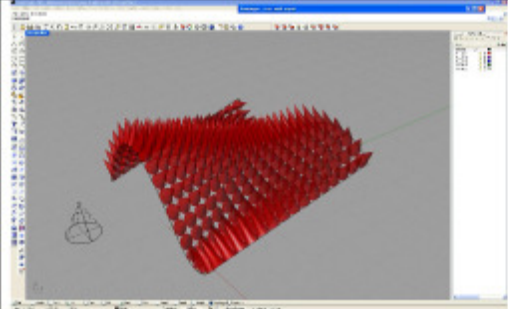
|  |  |  |   |
|--|--|--|---|
|  | <p><b>Creer_motif-morphing</b></p> <p>Répéter une B-Rep (existante) par « morphing » sur une surface (existante) divisée en fonction de U et V. Le morphing consiste à englober le motif dans une « boîte » et répéter cette boîte sur des volumes cibles positionnés sur la surface</p> | <p><i>Inputs :</i></p> <ul style="list-style-type: none"> <li>- B-Rep (motif)</li> <li>- Surface</li> <li>- Entier variable : V</li> <li>- Entier variable : U</li> </ul> <p><i>Output</i></p> <ul style="list-style-type: none"> <li>-Brep</li> </ul> | <p>Cours so823 séance 3</p> <p>creer_motif_morphing.ghx</p> |
|--|--|--|---|

Figure 220 : creer\_motif-morphing (catalogue So823-2011)

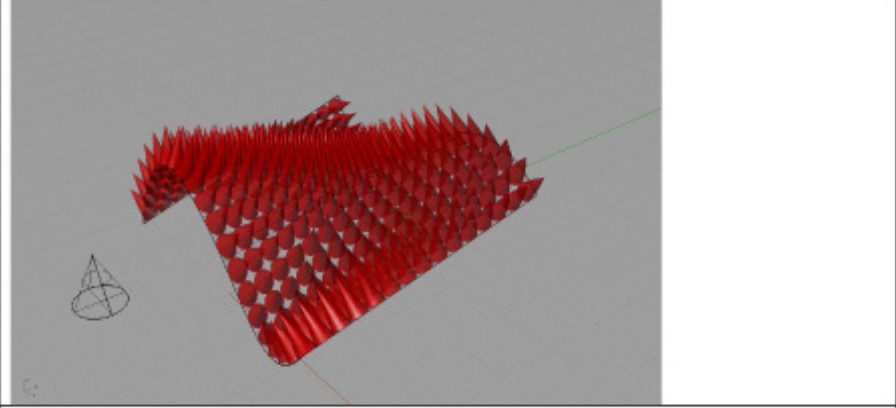
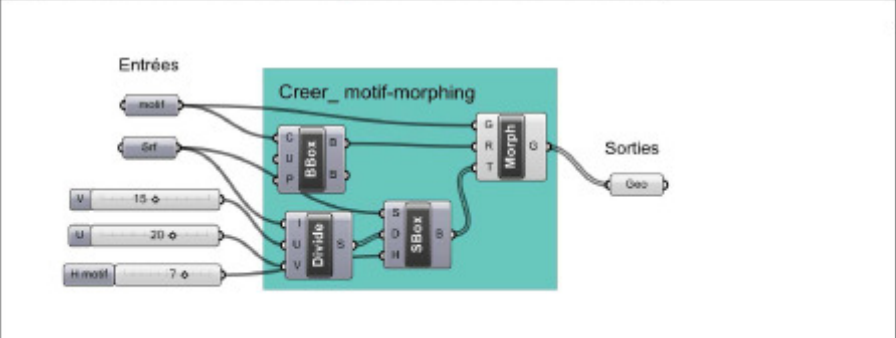
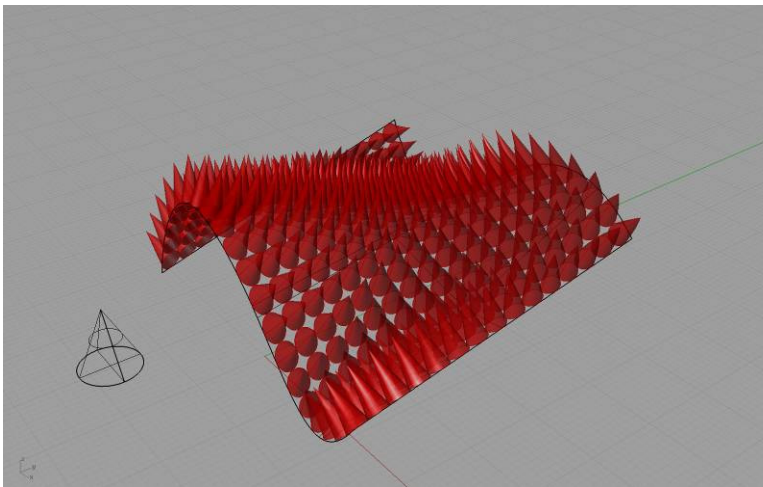
|  |   |  |
|--|---|--|
|  | <p><b>Creer_motif-morphing</b></p> <p>→ Crée un motif en répétant une géométrie (en volume et d'un bloc) sur une srf donnée en fonction de paramètres U et V</p> <p>Inputs :<br/>Motif (BRep d'un bloc), Srf, paramètres U et V</p> <p>Cf. SEANCE 3</p> | <p>GH v0.8.0006<br/>creer_motif_morphing.ghx</p> |
|  | <p>//////////</p>   |  |

Figure 221 : creer\_motif-morphing (catalogue So823-2012)

Les versions so823-2011, so823-2012 et DNArchi-2012 de ce sample sont identiques du point de vue du code.

Sample et inscription retenue dans le catalogue DNArchi-2012



**créer\_motif-morphing** →  
répéter un motif en répétant une géométrie (en volume et en un bloc) sur une srf donnée

GH v0.8.0006

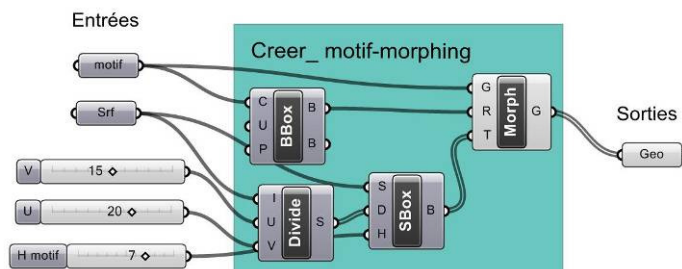
**créer\_motif-morphing**

Inputs :  
Srf, paramètres U et V définissant la trame, hauteur du motif

Outputs :  
Courbes

Cf. Articles  
« Tutoriel  
Morphing »

Exemples :



**Motif** **attracteur** **Reactor**

Figure 222 : créer\_motif-morphing (catalogue DNArchi-2012)

La version DNArchi-2012 associe au sample créer\_motif-morphing un renvoi vers l'article sur le pattern « Place Holder » (ainsi que vers un article de DNArchi « M comme Morphing » ?).

Les mots clefs choisis pour caractériser le sample créer\_motif-morphing sont :

- |   |                        |
|---|------------------------|
| - relatifs à la culture :                         | <b>Morphing</b>        |
| - relatifs aux composants et fonctions utilisés : | <b>Morphing</b>        |
| - relatifs aux notions sollicitées :              | <b>Motif, morphing</b> |
| - relatifs aux patterns impliqués :               | <b>Place Holder</b>    |
| - relatifs aux échelles dominant l'usage du       | <b>sans</b>            |



## Evaluer

### 3.21 evaluer\_planeite-surface

#### Spécificité du sample

Le sample evaluer\_planeite-surface a été utilisé lors du cours so823 pour illustrer la possibilité de réutiliser des codes existants diffusés en ligne. Plus spécifiquement, ce code est un des « samples » proposés par R. Woodbury et son équipe. Développés initialement sur Generative Components et diffusés sur <http://www.designpatterns.ca/>, ces codes ont été repris par Tsung-Hsien Wang sur Grasshopper et diffusés sur : <http://www.andrew.cmu.edu/org/tsunghsw-design/>. Ce sample exemplifie le pattern Reporter. Le pattern Reporter « Re-présente (abstrait ou transforme) des informations issues d'un modèle »<sup>121</sup> (Woodbury 2010, p.236).

Ce code, repris de celui proposé par Tsung-Hsien Wang, s'appuie principalement sur le composant « Planar » qui évalue la planéité d'une courbe, pour évaluer la courbure des sous-surfaces à partir de leurs contours.

#### Evolution du sample et de son inscription dans les catalogues proposés

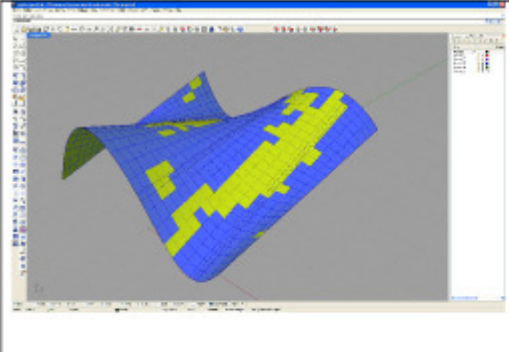
|   |   |  |  |
|---|---|--|--|
|  | <p><b>Evaluer_planeite-surface</b></p> <p>Evaluer la planéité d'une surface existante en fonction d'un delta (0=plan) et catégoriser les sous-surfaces en fonction d'un seuil</p> <p>source : <a href="http://www.andrew.cmu.edu/org/tsunghsw-design/">http://www.andrew.cmu.edu/org/tsunghsw-design/</a></p> | <p><b>Inputs :</b></p> <ul style="list-style-type: none"> <li>-Surface</li> <li>-Entier variable : V</li> <li>-Entier variable : U</li> <li>- Réel variable : seuil de planéité</li> </ul> <p><b>Outputs :</b></p> <ul style="list-style-type: none"> <li>-liste des « deltas »</li> <li>-listes des surfaces au-dessus et en-dessous du delta et visualisation de ces listes avec un code couleur (jaune = au dessus, bleu = au dessous)</li> </ul> | <p><a href="http://www.designpatterns.ca/">http://www.designpatterns.ca/</a></p> <p><a href="http://www.andrew.cmu.edu/org/tsunghsw-design/">http://www.andrew.cmu.edu/org/tsunghsw-design/</a></p> <p>Cours so823<br/>séance 10</p> <p>Evaluer_planeite-surface.ghx</p> |
|---|---|--|--|

Figure 223 : evaluer\_planeite-surface (catalogue so823-2011)

Ce sample étant déjà disponible en ligne et exemplifiant surtout l'utilisation d'un composant GH, il n'a pas été conservé dans les catalogues so823-2012 et DNArchi-2012.

### 3.22 Evaluer\_ecart-type

#### Spécificité du sample

Le sample evaluer\_ecart-type permet de calculer l'écart type d'une valeur, et est utilisé dans le cadre de la séquence pédagogique pour évaluer l'écart type entre les longueurs de barres d'une structure, ceci en vue d'être optimisé pour tendre à avoir des longueurs de barre toutes identiques.

<sup>121</sup> « Re-present (abstract or transform) information from a model » (Woodbury 2010, p.236).

Ce sample peut être rapproché du pattern Reporter (cf. infra 1.6.4).

Evolution du sample et de son inscription dans les catalogues proposés

|  |  |  |   |
|--|--|--|---|
| $S = \sqrt{\text{variance}} = \sqrt{\left(\frac{\sum(x - m)^2}{n - 1}\right)}$ | <b>Evaluer_ecart-type</b><br>Evaluer l'écart type d'une liste de valeurs, c'est-à-dire son | <i>Inputs :</i><br>-liste de réels ou d'entier<br><br><i>Outputs :</i> | Cours so823<br>séance 8<br><br>Evaluer_ecart- |
|--|--|--|---|

Figure 224 : evaluer\_ecart-type (catalogue so823-2011)

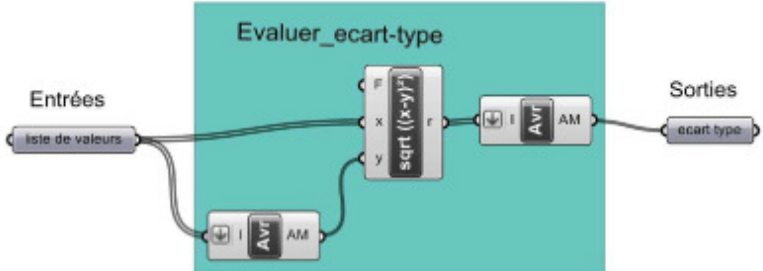
|   |  |   |
|---|--|---|
| $S = \sqrt{\text{variance}} = \sqrt{\left(\frac{\sum(x - m)^2}{n - 1}\right)}$  | <b>Evaluer_ecart-type</b><br>→ Evalue l'écart type d'une liste de valeur (c'est-à-dire son degré de dispersion par rapport à la moyenne)<br><br><i>Inputs :</i><br>Liste de valeurs<br><br><b>Cf. SEANCE 9</b><br><br>////////////// | GH v0.8.0006<br><b>Evaluer_ecart-type.ghx</b> |
|---|--|---|

Figure 225 : evaluer\_ecart-type (catalogue so823-2012)

Les versions so823-2011 et so823-2012 du sample sont identiques.

Sample et inscription retenue dans le catalogue DNArchi-2012

$$S = \sqrt{\text{variance}} = \sqrt{\left(\frac{\sum(x - m)^2}{n - 1}\right)}$$

**Evaluer\_ecart-type**  
 → évalue l'écart type d'une liste de valeur

*Inputs :*  
 Liste de valeurs (à plat)

*Outputs :*  
 Valeur de l'écart type

*Cf. Articles*

GH v0.8.0006  
**evaluer\_ecart-type**

Exemples :

**Evaluer Optimiser Reporter**

Figure 226 : evaluer\_ecart-type (catalogue DNArchi-2012)

Dans sa version DNArchi-2012, le sample evaluer\_ecart-type est associé au tutoriel « Waterloo » issu de la séquence pédagogique « Evaluation et Optimisation ».

|   |                                  |
|---|----------------------------------|
| <b>Les mots clefs choisis pour caractériser le sample evaluer_ecart-type sont :</b> |                                  |
| - relatifs à la culture :   | évaluer                          |
| - relatifs aux composants et fonctions utilisés :                                   | « fonctions »                    |
| - relatifs aux notions sollicitées :  | Evaluer et optimiser, écart type |
| - relatifs aux patterns impliqués :   | Reporter                         |
| - relatifs aux échelles dominant l'usage du sample :                                | sans                             |

### 3.23 Optimiser\_vers-cible

Evolution du sample et de son inscription dans les catalogues proposés

Ce sample n'a été intégré qu'à la version so823-2011 du catalogue. Il visait à exemplifier l'usage du composant « Galapagos » de Grasshopper. Cependant il semble trop spécifique à un composant pour être pertinent comme « sample ». Ce qui est intéressant ici semble plutôt être l'exemplification de l'usage de « Galapagos ». Dans les catalogue so823-2012 et DNArchi-2012, cela est intégré dans des « exemples », par exemple dans ceux liés au tutoriel « Waterloo ».

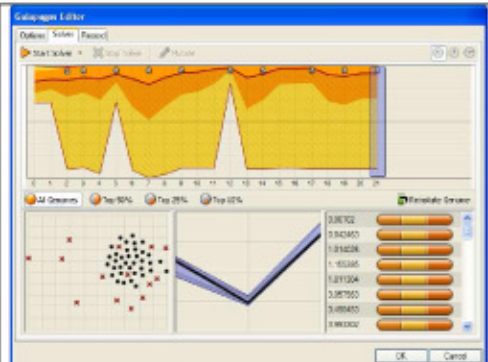
|   |   |   |   |
|---|---|---|---|
|  | <b>Optimiser_vers-cible</b><br>Permet de calculer l'écart avec une aire cible, cet écart pourra ensuite être minimisé par Galapagos | <b>Inputs :</b><br>- liste de réels<br>- réel « cible »<br><br><b>Output :</b><br>-écart à la cible, utilisable par Galapagos | Cours so823<br>séance 8<br><br>optimiser_vers-cible.ghx |
|---|---|---|---|

Figure 227 : optimiser\_vers-cible (catalogue so823-2011)

S'il était conservé, ce sample permettrait d'exemplifier le pattern Goal Seeker (Woodbury 2010, p.269).

### 3.24 Evaluer\_heliodon

#### Spécificité du sample

Le sample evaluer\_heliodon permet de créer un héliodon à partir de l'orientation du nord, de la hauteur du soleil à son azimuth et d'un temps t de la journée. Ce sample étant lié aux caractéristiques du site, on peut le rapprocher de l'échelle géographique.

Ce sample a été développé pour le projet d'Olivier Scheffer dans le cadre de son post-master à l'ENSCI. Il a ensuite été réutilisé dans le cours so823-2012 dans une séquence pédagogique visant à enseigner aux étudiants à évaluer l'ombre portée par un volume.

#### Evolution du sample et de son inscription dans les catalogues proposés

Ce sample a été soumis aux étudiants pour la première fois dans le cadre de l'enseignement so921-2011, lors de son appropriation des difficultés et remarques ont été émises par les étudiants : par exemple le code ne permettait pas encore de faire évoluer facilement le vecteur d'ensoleillement en fonction de l'heure de la journée. Ces remarques ont participé à la mise à jour du code pour le catalogue so823-2012.

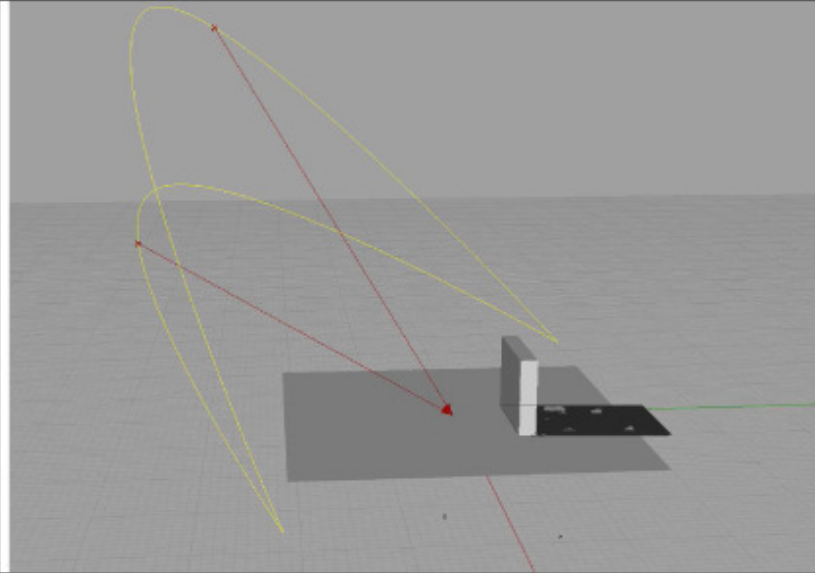
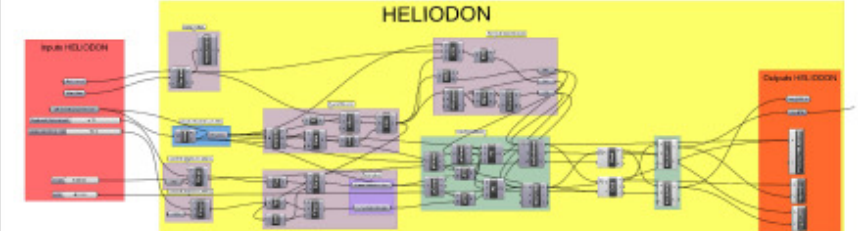

|   |  |  |
|---|--|--|
|   | <p><b>Evaluer-heliodon</b></p> <p>→ Crée un héliodon propre à un site</p> <p>Inputs :<br/>Emprise au sol du site choisi, orientation, angle du soleil à 12h : en hiver et en été, heure</p> <p>Outputs :<br/>Vecteur hivers et vecteur été</p> <p><b>Cf. SEANCE 10</b></p> | <p><b>GH v0.8.0006</b><br/><b>Heliodon.ghx</b></p> |
|  |    |  |

Figure 228 : evaluer\_heliodon (catalogue so823-2012)

#### Sample et inscription retenue dans le catalogue DNArchi-2012



**Evaluer\_ecart-type**  
→ Crée un héliodon propre à un site



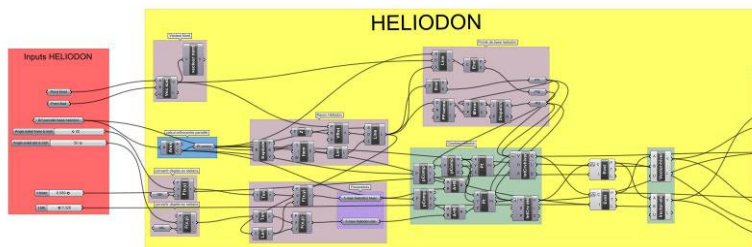
Inputs :  
Emprise au sol  
de la  
représentation de  
l'héliodon, angle  
du soleil à 12h,  
heure

GH v0.8.0006  
[evaluer\\_helidon.g  
hx](#)

Outputs :  
Vecteur d'hiver et  
vecteur d'été

Cf. Articles

Exemples :



**Evaluer ensoleillement « mesh shadow »**

**échelle(s) privilégiée(s) : échelle géographique ?**

**Les mots clefs choisis pour caractériser le sample evaluer\_helidon sont :**

- |   |                               |
|---|-------------------------------|
| - relatifs à la culture :                               | <b>Evaluer ensoleillement</b> |
| - relatifs aux composants et fonctions utilisés :       | <b>« mesh shadow »</b>        |
| - relatifs aux notions sollicitées :                    | <b>Mesh, vectors</b>          |
| - relatifs aux patterns impliqués :                     | <b>Projection ?</b>           |
| - relatifs aux échelles dominant l'usage du<br>sample : | <b>Echelle géographique</b>   |

## Récapitulatifs des samples proposés et de leurs caractéristiques

| Sample                      | C. du code | Ca t. | Pattern(s) associé(s) | Echelle associée   | Mots clefs   |
|-----------------------------|------------|-------|-----------------------|--------------------|--|
| Extrait_1item_liste4        | F          | 1     | Place Holder          |                    | Motif, set, arbre de données, Place Holder   |
| Mapping_list                | F          | 1     | Selector              |                    | Mapping, Arbre de données  |
| Extrait_valeur-max-min      | F          | 1     | Selector              |                    | Sort List, Arbre de données, Selector  |
| creer-trame-pts_UV          | F          | 2     | Points collection     |                    | Motif, trame, Point, sous-surface, Point Collection, Place Holder  |
| Creer_pt_UV                 | F          | 2     | Reactor               |                    | UV, Eval Pt on Srf, Reactor, Controller  |
| creer_liste-sous-srf_UV     | F          | 2 ?   | Place Holder          |                    | Motif, sous-surface, Composant, Place Holder   |
| creer_pts_aleatoires        | F          | 2     | Controller            |                    | Random, Point  |
| Creer_crv_math_spirale      | E          | 3     | Points collection     | Modèle géométrique | Spirale, fonction, Courbes, Point, équation paramétrique, Point Collection, éch. de modèle, éch. géométrique |
| Creer_crv_math_spirale3d    | E          | 3     | Point collection      | Modèle géométrique | Spirale, fonction, Courbes, Point, équation paramétrique, Point Collection, éch. de modèle, éch. géométrique |
| Creer_crv_math-balle-tennis | E          | 3     | Point collection      | Modèle géométrique | fonction, Courbes, Point, équation paramétrique, Point Collection, éch. de                                   |

|                                |   |   |                                  |              |   |
|--------------------------------|---|---|----------------------------------|--------------|---|
|                                |   |   |                                  |              | modèle, éch. géométrique  |
| Creer_crv_sections-srf         | E | 3 | sans                             |              | Section, Intersection, Courbe   |
| Creer_crv_RandomVoronoi        | E | 3 | sans                             | modèle       | Voronoi, courbe, éch. de Modèle   |
| creer_maillage_random-delaunay | E | 4 | Point Collection                 | parcellaire  | Delaunay, Random, Mesh, Point Collection, éch. parcellaire  |
| creer_srf_repulseur            | E | 4 | Reactor Point Collection mapping | parcellaire  | Reactor, srf, mapping, point collection, éch. Parcellaire, attracteur   |
| Creer_srf_sections             | E | 4 | sans                             |              | Evaluer, Surfaces constructibles, Intersections,  |
| Creer_srf_mathShell            | E | 4 | Point Collection                 | Modèle?      | Coquillage, fonction, surface, Point, équation paramétrique, Point Collection, éch. de modèle, éch. géométrique |
| creer_motif-maillage           | E | 5 | Place Holder                     | technique    | Motif, composant, Place Holder, éch. technique  |
| Creer_motif-reactif_attracteur | E | 5 | Reactor Place Holder Mapping     |              | Mapping, Motif, Reactor, attracteur   |
| Creer_motif-morphing           | E | 5 | Reactor Place Holder             |              | Morphing, Motif, Place Holder   |
| Evaluer_ecart-type             | F | 6 | Reporter                         |              | Evaluer, fonction, reporter, écart type   |
| Optimiser_vers-cible ?         | E | 6 | Goal Seeker                      |              | Goal Seeker, evaluer, optimiser, galapagos  |
| Evaluer_heliodon               | F | 6 | Projection                       | géographique | Evaluer, ensoleillement, vector, Projection   |
| evaluer_verticale-max          | F | 6 | Selector                         |              |   |

**Légende :**

**Catégorie**

1 = Organiser des arbres de données

2 = Créer des points

3 = Créer des courbes

- 4 = Créer des surfaces
- 5 = Créer des motifs
- 6 = Evaluer

### **Complexité**

- F = Fragment
- E = Exemple

**Figure 229 : Synthèse des caractéristiques des samples construits pour DNArchi-2012**

## **3.25 Récapitulatif des évolutions des catalogues de samples construits**

L'évolution des samples est marquée par trois étapes : -la mise en place du catalogue pour l'Expe1, -la modification du catalogue suite à l'Expe1 et son évaluation lors des Expe2 et 3, et enfin – la mise en place de la plateforme [parametric-ressources].

### **- Support de Samples version 1.1**

Les samples que nous présentons ici ont été initiés dans le cadre de l'enseignement So 823 de l'ENSA Paris la Villette lors des semestres 2 de l'année 2010-2011 et 2011-2012, correspondant à un enseignement de Grasshopper en Master 2 (24h par semestre). Ces fichiers ont été construits en mai 2011 pour formaliser des notions du cours abordées lors du semestre. Les samples, alors au nombre de vingt et un, étaient mis à disposition des étudiants *via* le portail de l'enseignement en ligne. Les samples étaient publiés sous la forme d'un catalogue, structuré en catégories. Ces catégories sont liées aux « sorties » des codes, c'est-à-dire à ce qu'ils permettent de faire : -organiser des arbres de données ; -créer des points, -créer des courbes, -créer des surfaces, - créer un motif, et – évaluer. Cette organisation par catégorie est plus basique et plus directement accessible pour des non-experts de la modélisation paramétrique que la classification par Patterns proposée par R. Woodbury (Woodbury 2010). La classification des Samples par Patterns peut néanmoins se retrouver dans la version finale de la diffusion des samples, sur [parametric-ressources]. Dans sa version1, le catalogue comprenait une image illustrative du fragment de modèle, un commentaire sur la manière dont opère le modèle, une description des entrées et des sorties ainsi qu'un renvoi vers le cours concerné par la notion sollicitée (*cf.* Figure 230).

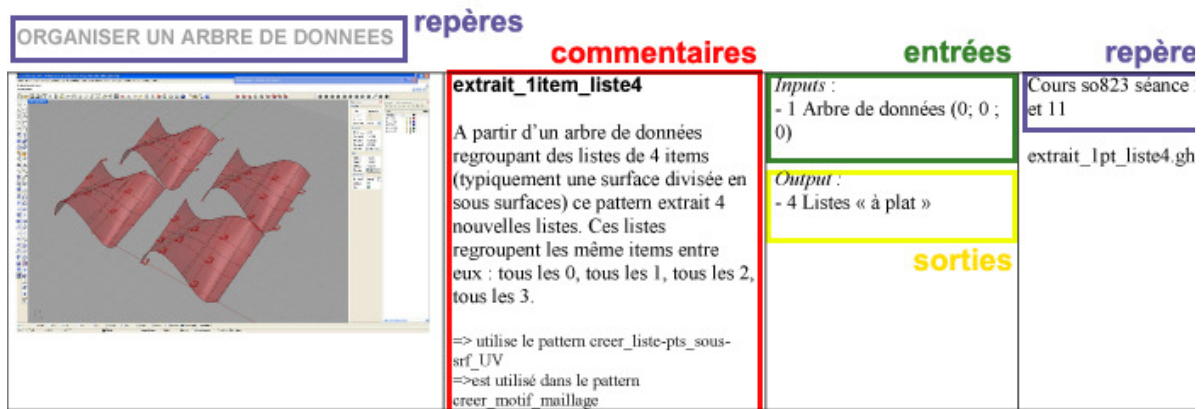


Figure 230: Organisation de la bibliothèque de Patterns mise à disposition des étudiants (version0.1)

Les modèles eux-mêmes ont été structurés de manière spécifique, en vue d'être lisibles et de fournir un maximum d'informations aux étudiants. Commentaires, organisation par groupe, titres et explicitations des entrées et des sorties ont été mis en avant dans chacun des modèles diffusés (cf. Figure 231). Par exemple dans le Pattern « extrait\_1pt\_liste4 » (cf. Figure 231) les entrées sont rendues lisibles par leur explicitation au travers de l'utilisation d'un composant spécifique (cf. Figure 231-A). Des commentaires et une organisation des étapes du modèle par « groupes » (cf. Figure 231-B) guident la lecture du script. Des composants explicitent également les différentes sorties du modèle pouvant être intéressantes (cf. Figure 231-C). Les composants explicitant les entrées et les sorties du modèle ne participent pas à la génération d'une géométrie, ces éléments ne participent qu'à la lisibilité du modèle, ils pourraient être supprimés sans modifier la géométrie créée.

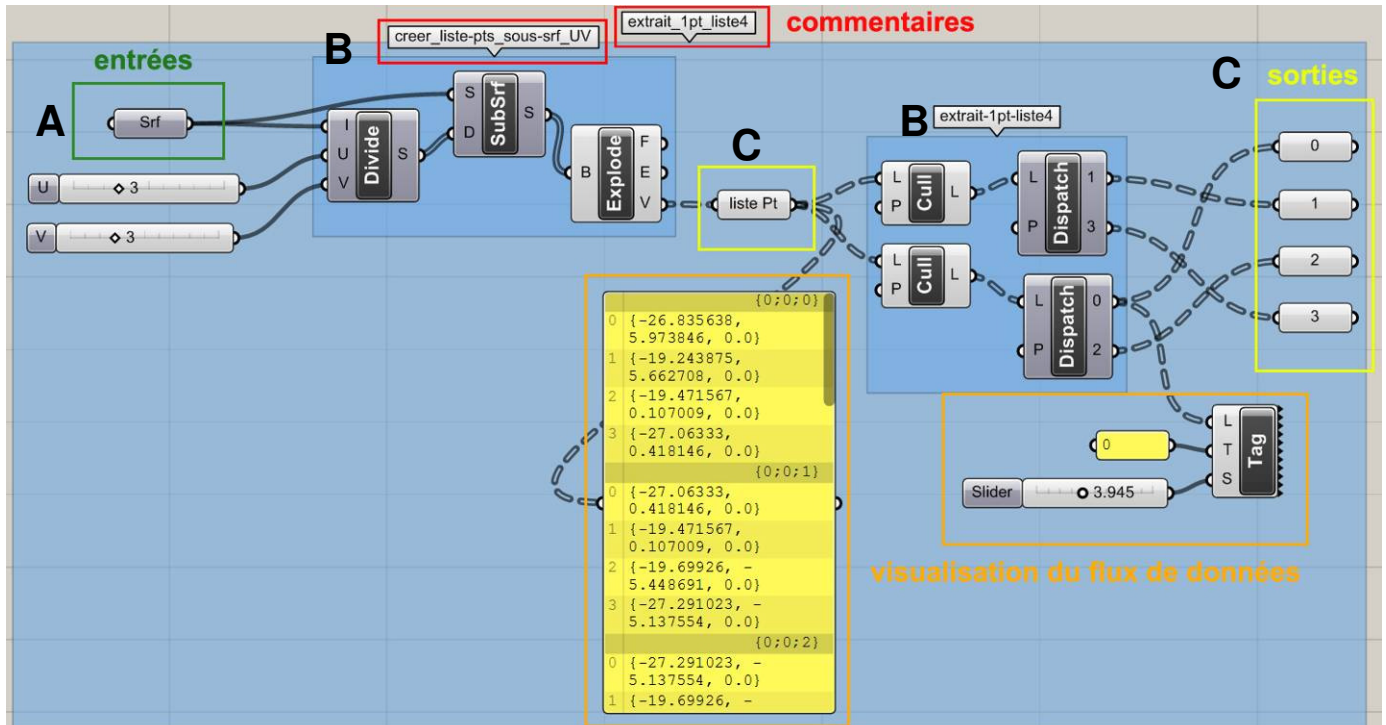


Figure 231 : Modèle Grasshopper mis en forme comme Sample

Lors de l'Expe1, il a été plus difficile que prévu pour les étudiants d'entrer dans des fichiers qui n'étaient pas les leurs ou qui ne leur étaient pas explicitement décrits. Cependant les étudiants ont fait preuve de curiosité et ont exploré les ressources fournies. L'analyse de l'usage des samples par les

étudiants lors de l'Expe1 (cf. Chapitre 8) montre que les supports de cours étaient très utilisés : les étudiants, même s'il leur était demandé d'utiliser des fichiers préconstruits, préféraient souvent reconstruire les fragments de modèles eux-mêmes à partir des indications du cours. Les images montrant les modèles eux-mêmes (les agencements des composants plutôt que l'image de la géométrie produite) étaient alors très utilisées. Ce résultat à été mis à profit pour mettre à jour le catalogue dans sa version 2. Une fois dépassées les premières difficultés de compréhension, les modèles eux-mêmes ont convenu. Ils ont pu être réappropriés par les étudiants (cf. Chapitre 8).

### - Support de samples, Version 1.2

A la description des samples dans la bibliothèque a été intégrer une image permettant de visualiser le modèle sans avoir à ouvrir le fichier (cf. Figure 231A). Les textes de présentation des samples ayant très peu été lus, ils ont été revus et parfois simplifié pour en rendre explicite les mots clefs (cf. Figure 231B). Les noms des Samples ont également été modifiés pour être plus lisibles et fournir des informations sur le contenu du modèle. La description des entrées et des sorties du modèle a été gardé par rapport au catalogue version 1, ainsi que l'explicitation de la référence au cours, celle-ci ayant été très utilisée lors de l'Expe1.

## A- image du modèle

## B- description

## C- exemples

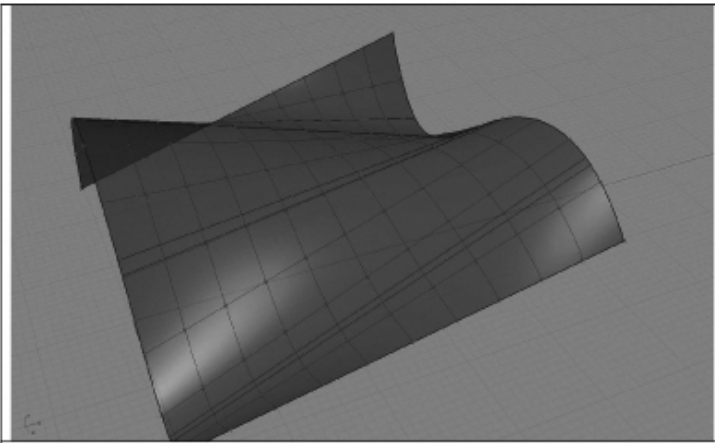
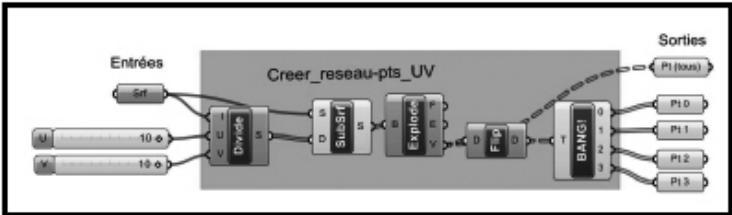
|   |   |  |
|---|---|--|
|  | <p><b>Créer_trame-pts_UV</b></p> <p>→ Crée une trame de point sur une surface (a 4 côtés) selon deux divisions (U et V)</p> <p>Inputs :<br/>Surface, Paramètres U et V</p> <p>Outputs :<br/>une trame de points</p> <p style="border: 1px solid black; display: inline-block; padding: 2px;">Cf. SEANCE 2</p> | <p>GH v0.8.0006<br/><b>Créer_trame-pts_UV.ghx</b></p>  |
|  | <p><b>Exemples :</b></p> <p>// Créer une trame de pts sur une surface pour y « populer » un motif</p> <p>// Créer une trame de pts régulière et supprimer certains pts aléatoirement</p>  | <p>GH v0.8.0006<br/><b>creer_motif_maillage.ghx</b></p> <p>GH v0.8.0006<br/><b>Créer_pts-aleatoires_sur-trame-UV.ghx</b></p> |

Figure 232 : Construction du catalogue version1.2

La formalisation des commentaires à également été mise à jour : dans la version 1 les commentaires étaient compris dans les noms de groupes et, lorsque les groupes étaient modifiés les noms étaient perdus. Des commentaires sous la forme de textes ont donc plutôt été mis en place (cf. Figure 232-A). Dans les cas des modèles relativement étendus ou pouvant demander des modifications

spécifiques, plus poussées que la manipulation d'un slider, des commentaires plus longs ont été introduits. Par exemple dans le sample créer\_motif-reactif-attracteur.ghx, des commentaires (sous forme de « panels ») donnent des précisions quant à la nature de l'entrée et aux possibles variations.

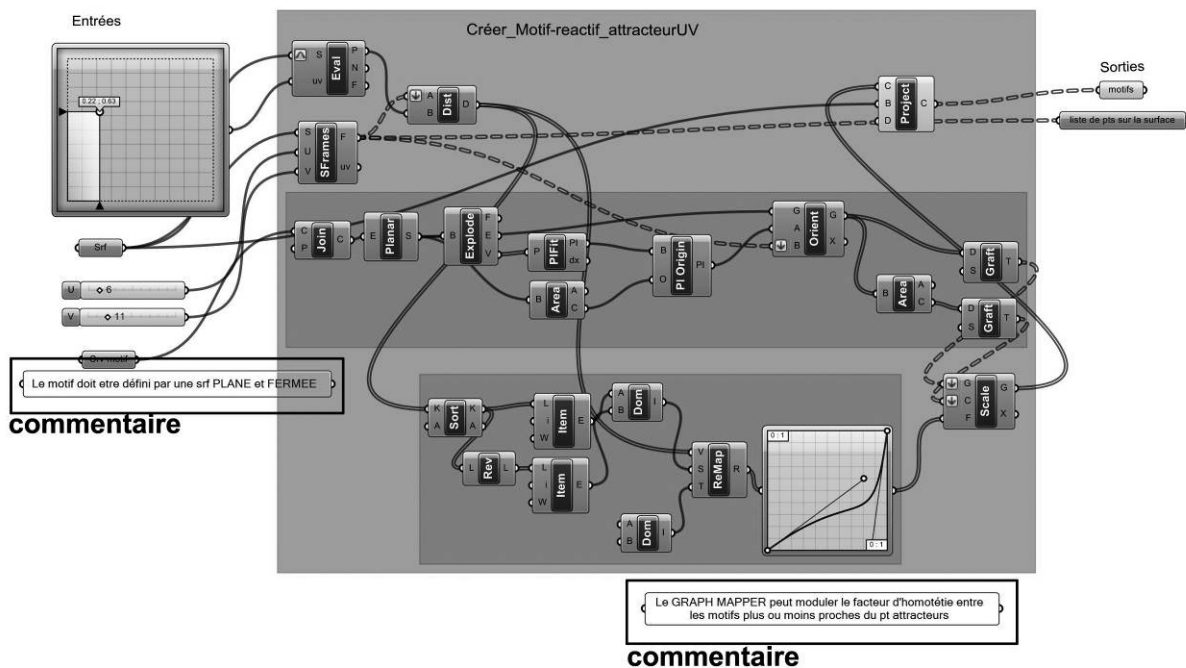


Figure 233 : modèle GH du sample créer\_motif-reactif-attracteur.ghx version 1.2

Si la majorité des samples du catalogue version 1 ont convenu, d'autres modèles ont été développés et certains ont été modifiés pour être plus généraux par rapport aux notions en cours. Ces modifications ont parfois introduit des complexités supplémentaires, mais elles ouvrent le champ des possibles. Par exemple pour le sample créer\_motif-reactif-attracteur.ghx, le motif qui initialement était un hexagone « par défaut », a été modifié pour pouvoir être n'importe quelle courbe plane et fermée. Les évolutions du catalogue, sample par samples sont explicitées dans l'Annexe 3 p.417.

### - Support de samples version 3 : [parametric-ressources]

La plateforme [parametric-ressources] a été construite en janvier 2013 suite aux analyses des Expe2 et 3 et, a été utilisée et évaluée lors de l'Expe 4 ainsi que lors d'un enseignement de projet à l'ENSA de Versailles (le Studio SP31, sous la responsabilité de Claire Petetin). La plateforme [parametric-ressources] vise à : - rendre les samples visibles, lisibles et accessible à tous, et – inscrire les sample dans un « espace des savoirs de la modélisation paramétrique » grâce au ressources du web, en mettant en place des liens (liens hypertextes, renvois) vers des articles, des sources et des ressources diverses.

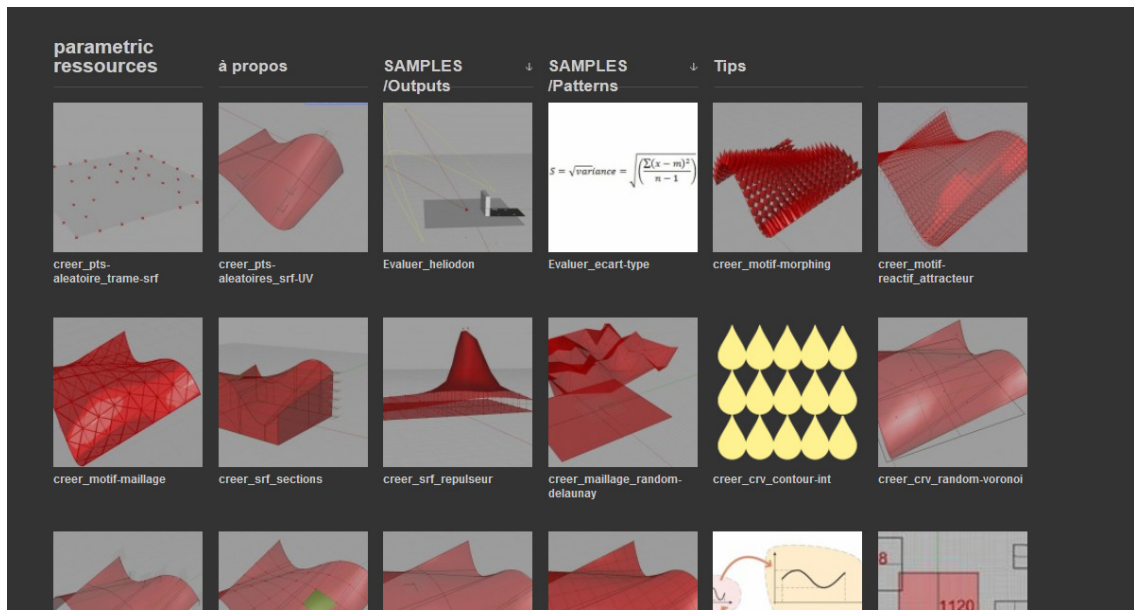


Figure 234 : page d'accueil de [parametric-ressources]



## ANNEXE 4 :

# RETOURS SUR ENSEIGNEMENTS

## 4 Retour sur les enseignements so823-2011 et so823-2012

### 4.1 Enseignement so823-2011 : description et retour critique

| <i>Date</i>            | <i>Contenu de la séance</i>   | <i>Thème de la séquence</i>                         |
|------------------------|---|---|
| Séance 1<br>2011-03-03 | Introduction à Grasshopper. Exemple du Gerkin, début de l'exercice « city hall »  | introduction à Grasshopper et aux arbres de données |
| Séance 2<br>2011-03-10 | Présentations des tableaux en programmation, division d'une surface selon deux paramètres (u et v)<br>Création d'un maillage à partir de la trame de point<br>Fin de l'exercice « City Hall » |   |
| Séance 3<br>2011-03-17 | Répétition d'un motif contenu dans une « boîte enveloppante » sur des « boîtes cibles » instanciées sur une surface<br>Exemple/Exercices du tissage <i>cf</i> NOX                             | Morphing  |
| Séance 4<br>2011-03-24 | Création d'objets en 2D ou 3D à partir de fonctions<br>Génération aléatoire<br><i>Exercice</i> : musée de Doa : répéter des objets (disques circulaire) de façon                              | Programmer courbes et surfaces à                    |

|                         |  |  |
|-------------------------|--|--|
|                         | aléatoire sur une courbe défini par une fonction<br>Utiliser une courbe à partir d'une fonction pour faire un « morphing »   | partir d'équations paramétriques       |
| Séance 5<br>2011-03-31  | Corrigé de l'exercice sur le musée de Doa (expérimentation sur les listes de points obtenus => comment modifier les listes pour les traiter différemment (ex tous les points ayant le même u ou le même v, etc)        |  |
| Séance 6<br>2011-04-07  | Attracteurs / paroi orientable<br>Contrôle d'un objet ou groupe d'objet grâce à sa position relative à un tiers  | Attracteurs                            |
| Séance 7<br>2011-04-28  | Triangulation de Delaunay et diagramme de Voronoi  | Voronoi et Delaunay                    |
| Séance 8<br>2011-05-05  | Evaluation et Optimisation : Utilisation de Galapagos pour optimiser l'aire totale des surfaces constructibles du modèle « Gherkin ». Exercice pour optimiser certains des éléments structurels de la gare de Waterloo | Evaluation et Optimisation             |
| Séance 9<br>2011-05-12  | Evaluation et Optimisation (suite)<br>Correction de l'exercice sur Waterloo et travail sur une évaluation des courbures de surface   |  |
| Séance 10<br>2011-05-19 | Introduction à la méthode des Patterns, exercice avec utilisation de codes fournis.  | Introduction à la méthode des patterns |
| Séance 11<br>2011-05-26 | Examen : TP piscine, groupes A   |  |
| Séance 12<br>2011-06-09 | Examen : TP piscine, groupes B   |  |

## **4.2 Enseignement so921-2011 : description et retour critique**

Ce cours de scripting visait à enseigner aux étudiants à se programmer leurs propres composants sur Grasshopper en VB. Néanmoins la quasi-totalité des étudiants n'ayant aucune connaissance en Grasshopper, la première partie du cours (séance 1 à 6) a repris l'introduction à Grasshopper construite en so823-2011.

|                         |   |  |  |
|-------------------------|---|--|--|
| Séance 1<br>2011-09-29  | Idem séance 1-so823-2011  | introduction à Grasshopper et aux arbres de données                    |  |
| Séance 2<br>2011-10-06  | Idem séance 2-so823-2011  |  |  |
| Séance 3<br>2011-10-13  | Idem séance 3-so823-2011  | Morphing   |  |
| Séance 4<br>2011-10-20  | Idem séance 6-so823-2011  | Attracteurs  |  |
| Séance 5<br>2011-10-27  | Idem séance 9-so823-2011  | Evaluation et Optimisation   |  |
| Séance 6<br>2011-11-03  | Idem séance 10-so823-2011   |  |  |
| Séance 7<br>2011-11-10  | Introduction à la programmation de nouveaux composants Grasshopper : type de données, structures de contrôle , notion de boucle | Introduction à la programmation de nouveaux composants sur Grasshopper |  |
| Séance 8<br>2011-11-17  | Développement d'un algorithme pour évaluer la courbure d'une courbe   | Evaluer la courbure de courbes et de surfaces                          |  |
| Séance 9<br>2011-11-24  | Suite de l'algorithme pour évaluer la courbure d'une surface  |  |  |
| Séance 10<br>2011-12-01 | Introduction à la méthode des patterns.<br>Exercice du moucharabié.   |  |  |
| Séance 11<br>2011-12-08 | Examen : TP piscine   |  |  |


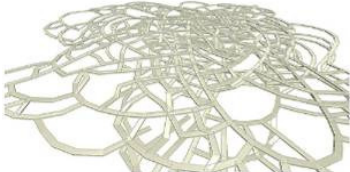
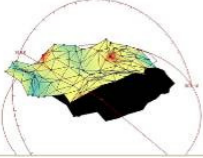
### 4.3 Enseignement so823-2012 : description et retour critique

| Date                   | Contenu de la séance  | Thème de la séquence                                | Ressources correspondantes  |
|------------------------|---|---|---|
| Séance 1<br>2012-03-01 | Idem séance1-so823-2011   | introduction à Grasshopper et aux arbres de données | → créer-maillage<br>→ extrait litem   |
| Séance 2<br>2012-03-08 | Idem séance2-so823-2011   |   |   |
| Séance 3<br>2012-03-15 | Idem séance3-so823-2011   | Morphing  | → morphing  |
| Séance 4<br>2012-03-22 | Exercice de tissage, manipulation d'une liste de points et transformations de points  | Tissage   | → référence au tutoriel « Generativ Algorithms, Weaving » de Zubin Khabazi<br><a href="http://www.grasshopper3d.com/page/tutorials-1">http://www.grasshopper3d.com/page/tutorials-1</a> |
| Séance 5<br>2012-03-29 | Approfondissement des arbres de données, manipulation et visualisation des données, transformation de données grâce au « Graph Mapper » | Approfondissement des arbres de données             |   |
| Séance 6<br>2012-04-05 | Triangulation de Delaunay et diagramme de Voronoi, transformation dans la génération des points par rapport aux exercices so823-2011    | Voronoi et Delaunay                                 | →   |
| Séance 7<br>2012-04-12 | Idem séance6-so823-2011   | Attracteurs   | → tutoriel attracteurs  |
| Séance 8<br>2012-05-03 | Idem séance4 et 5-so823-2011  | Programmer courbes et surfaces à partir d'équations | →   |

|                         |  | paramétriques                          |   |
|-------------------------|--|--|---|
| Séance 8<br>2012-05-10  | Idem séance8-so823-2011  | Evaluation et Optimisation             | → tutoriel Waterloo<br>→ tutoriel évaluer l'enseiillement |
| Séance 9<br>2012-05-24  | Idem séance9-so823-2011  |  |   |
| Séance 10<br>2012-05-31 | Evaluer l'enseiillement : utiliser le composant « MeshShadow » avec le sample Héliodon. Exercice de moucharabié. |  |   |
| Séance 11<br>2012-06-07 | Introduction à la « méthode des patterns »<br>Suite de l'exercice du moucharabié.                                | Introduction à la méthode des patterns |   |
| Séance 12<br>2012-06-14 | Examen : TP piscine  |  |   |

**ANNEXE 5 :**  
**Analyse des aides à la**  
**modélisation**  
**paramétrique disponibles**  
**en ligne**

## 5 Supports en ligne d'aide à la modélisation paramétrique

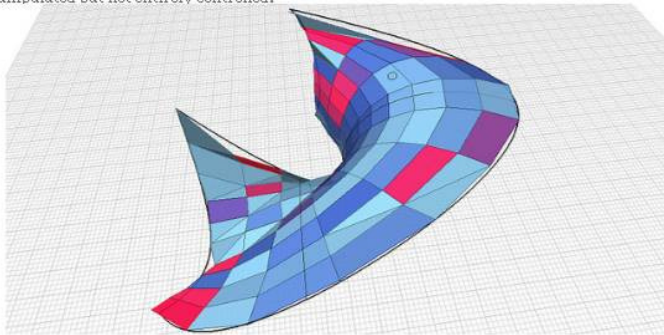
|  |  |
|--|--|
| <p>GRASSHOPPER</p> <hr/> <p>AGGREGATION:</p>  <p><i>Diffusion Limited Aggregation</i></p>  <p><i>Auxin Flux Canalisation</i></p> <p>ANALYSIS:</p>  | <p>→ Liste de modèles organisés par catégories et décrits uniquement par un titre et une image</p> |
|--|--|

**PO HYBRID**

Posted by [trevor.patt](#) on 06/06/2011 - [Leave a Comment](#)

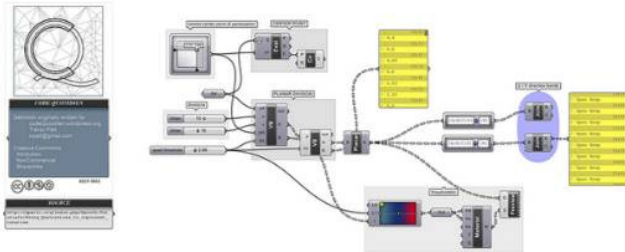


Discretization of complex surfaces into continuous planar quadrilaterals is not a terribly complex process; however, it is very sensitive to small adjustments in the initial conditions, so optimizing the panelization *is*. The simplest solution is to iterate many, many times over the entire mesh searching for an *equilibrium*. This definition is not concerned with optimization but takes advantage of parametric visualization to allow the user to adjust the initial conditions and observe the difference these small changes make. In this way, some of the irregularity of the unoptimized discretization is preserved and provides resistance to the smooth surface input in a way that can be manipulated but not entirely controlled.



The VB.NET script works from an origin point outward across a surface in quadrants, so the seams which pass through the origin will all be on the surface with increasing divergence toward the corners. This means that positioning the origin on an edge ensures that the discretized edge is consistent with the surface's edge. Should the quadrilateral panelization diverge too much from base surface, the threshold slider sets a limit (in absolute distance from the surface) at which a quadrilateral will be split into two triangles to force realignment. Setting the threshold to 0.0 obviously produces an entirely triangular panelization.

[\(ghx download\)](#)



→ Description des applications possibles

→ Description du modèle produit

→ Description des mécanismes de la définition Grasshopper

*Discretization of complex surfaces into continuous planar quadrilaterals is not a terribly complex process; however, it is very sensitive to small adjustments in the initial conditions, so optimizing the panelization [is](#). The simplest solution is to iterate many, many times over the entire mesh searching for an [equilibrium](#). This definition is not concerned with optimization but takes advantage of parametric visualization to allow the user to adjust the initial*

→ Rappel des objectifs du code

→ Description des enjeux (ici de la



|  |  |
|--|--|
| <p><i>conditions and observe the difference these small changes make. In this way, some of the irregularity of the unoptimized discretization is preserved and provides resistance to the smooth surface input in a way that can be manipulated but not entirely controlled.</i></p> <p><i>The VB.NET script works from an origin point outward across a surface in quadrants, so the seams which pass through the origin will all be on the surface with increasing divergence toward the corners. This means that positioning the origin on an edge ensures that the discretized edge is consistent with the surface's edge. Should the quadrilateral panelization diverge too much from base surface, the threshold slider sets a limit (in absolute distance from the surface) at which a quadrilateral will be split into two triangles to force realignment. Setting the threshold to 0.0 obviously produces an entirely triangular panelization.</i></p> <p><a href="#"><i>(ghx download)</i></a></p> | <p>pannelisation)</p> <p>→ Description du fonctionnement du code</p> |
|--|--|

**Tableau 46 : code quotidien » (capture d'écran de <http://codequotidien.wordpress.com/grasshopper/> le 20-03-2012)**

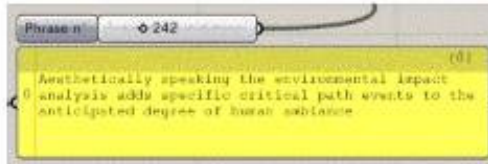


. A simple GH definition that mimics the Escher patterns process.

[Seamless pattern - Co-de-IT - GH080066.zip](#) < Grasshopper definition (for GH v0.8.0066)

---

DIY Architectural Dialogue machine.



. Inspired by this post - <http://allsouarchitecture.tumblr.com/post/5613643435>

I did a GH def to sort random strings from several lists of data in order to do an architectural dialogue machine.

[DIY Arch Dialogue.zip](#) < Grasshopper definition (for GH v0.8.0010)

---

Blender2GH.



a simple GH definition to import a Blender 3D geometry using the export.PLY file format

[Blender2GH\\_blander.zip](#) < Blender 3D file (for Blender 3D 2.57b)

[Blender2GH\\_PLY.zip](#) < export.PLY file

[Blender2GH\\_GH\\_def.zip](#) < Grasshopper definition (for GH v0.8.0010)

[Blender2GH\\_tutorial.zip](#) < Tutorial (PDF file)

---

→ Liste de modèles organisés décrits par un titre, une image et quelques mots.

→ les versions des fichiers sont indiquées, différentes versions sont proposées


Tableau 47 : Capture d'écran de « co-de-it » (le 20-03-2012)

→ galerie, tutoriels, forum, événement, espace personnel

→ visibilité des membres inscrits

→ visibilité des dernières discussions postées sur le forum

Tableau 48 : Page d'accueil de grasshopper3d.com (capture d'écran de grasshopper3d.com le 19-03-2012)



### Points on surface

Posted by [Milan Katić](#) on March 18, 2012 at 3:24pm in [Discussion](#)  
[View Discussions](#)

I tried to find deffinition here but i couldn't.  
 What i want to do is to put random points on surface.  
 First i tried with freeform cloud, but i dont want to have **groups of points that stick to edge** of surface. I found some deffinition here, but it not solve the problem..  
 I would like to get some deffinition with random points INSIDE surface,  
 or some witch have better deployed points.

Please ii hope you have some ides for this.  
**Or anything that can make better points on surface than freeform cloud.**  
 Thanks!!


[Share](#)
[Twitter](#)
[Facebook](#)

Views: **75**

[▶ Reply to This](#)

---

**Replies to This Discussion**




[Reply by \[Giulio Piacentino\]\(#\) 15 hours ago](#)

Hi Milan,  
 do you mean inside like "in" the volume enclosed by the surface or like "on" the thin skin of the surface?  
 - Giulio

[giulio@mcneel.com](mailto:giulio@mcneel.com)

[▶ Reply](#)

---




[Reply by \[Alberto Mora\]\(#\) 15 hours ago](#)

Hi Milan,  
 Try with this.  
 It's a geometric solution to random points ON any surface.  
 Alberto

Attachments:  
[random points on surface\\_AMB.ghx](#), 199 KB

[▶ Reply](#)

**Figure 235 : Extrait d'une discussion "Samples and Examples" (capture d'écran de grasshopper3d.com le 19-03-2012)**



### Weaving Cloth

Posted by [Kenneth cheung](#) on December 30, 2011 at 1:46am in [Sample and example files](#)  
[View Discussions](#)

Hi All.

This is my first post in the discussion forum. If there are any bugs or any errors in the gh definition, please feel free to comment on them.

The difficult part of the script is the order of the points,path mapper and other list issues are the greatest challenges.

The original idea was inspired by **Generative Algorithms\_CaE\_Weaving** by **Zubin Khabazi**.



Figure 236 : Exemple de discussion "Samples and Examples" (capture d'écran de grasshopper3d.com le 19-03-2012)

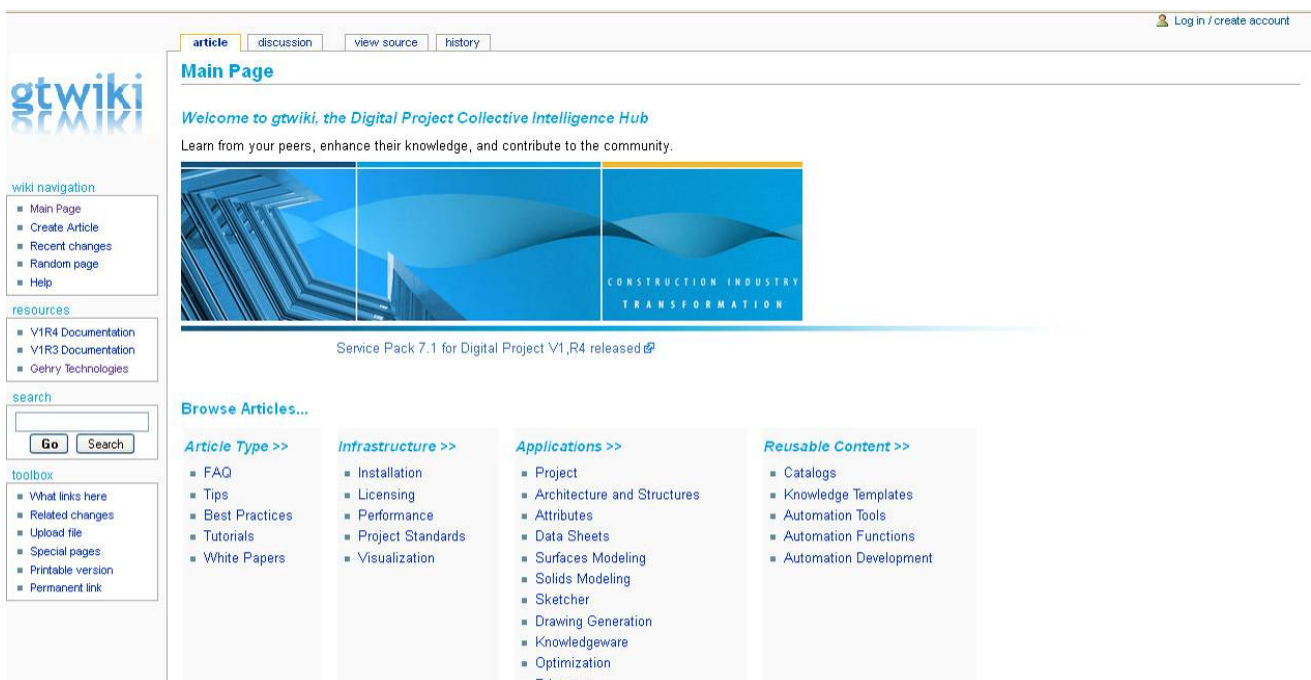
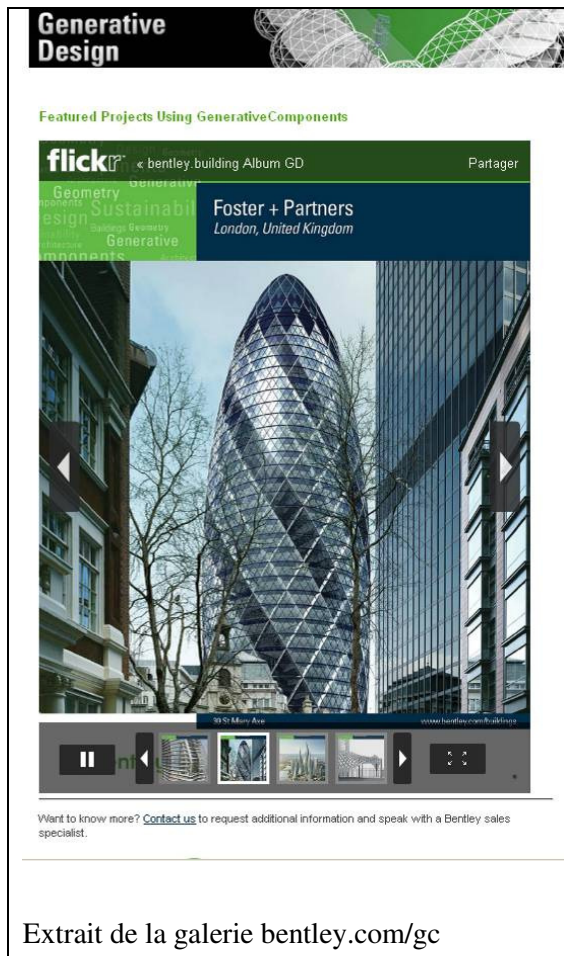


Figure 237 : Page d'accueil de gtwiki.com (capture d'écran de gtwiki.com le 19-03-2012)





Extrait de la galerie bentley.com/gc



Extrait de la galerie grasshopper3d.com

Tableau 49: Cas de mise en scène de projets réalisés, sur bentley.com/gc et grasshopper3d.com (captures d'écran du 02-04-2012)

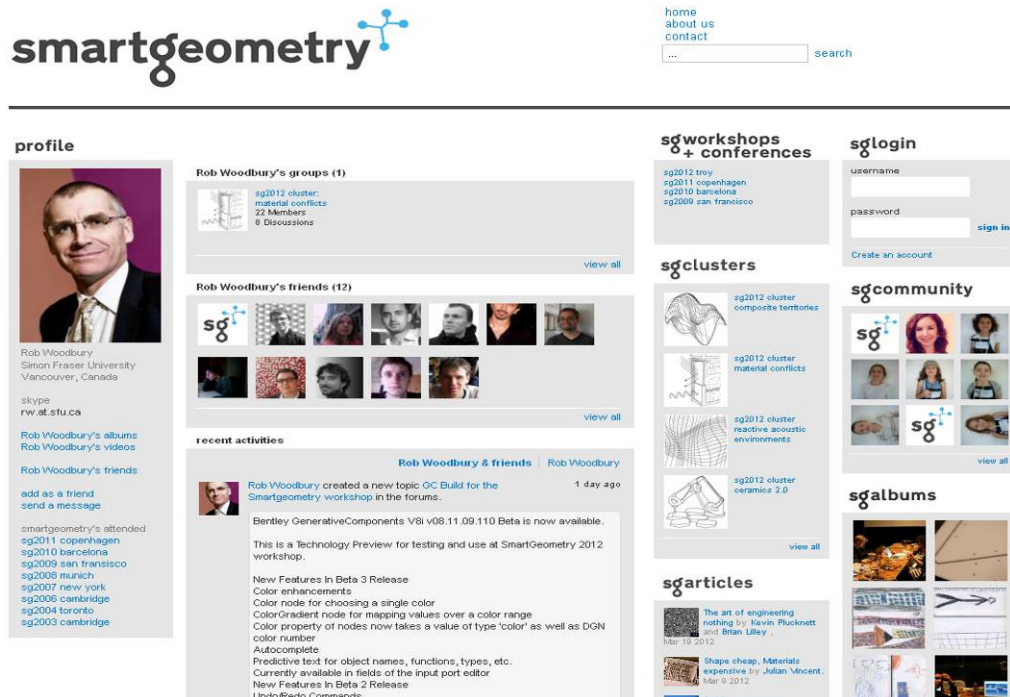
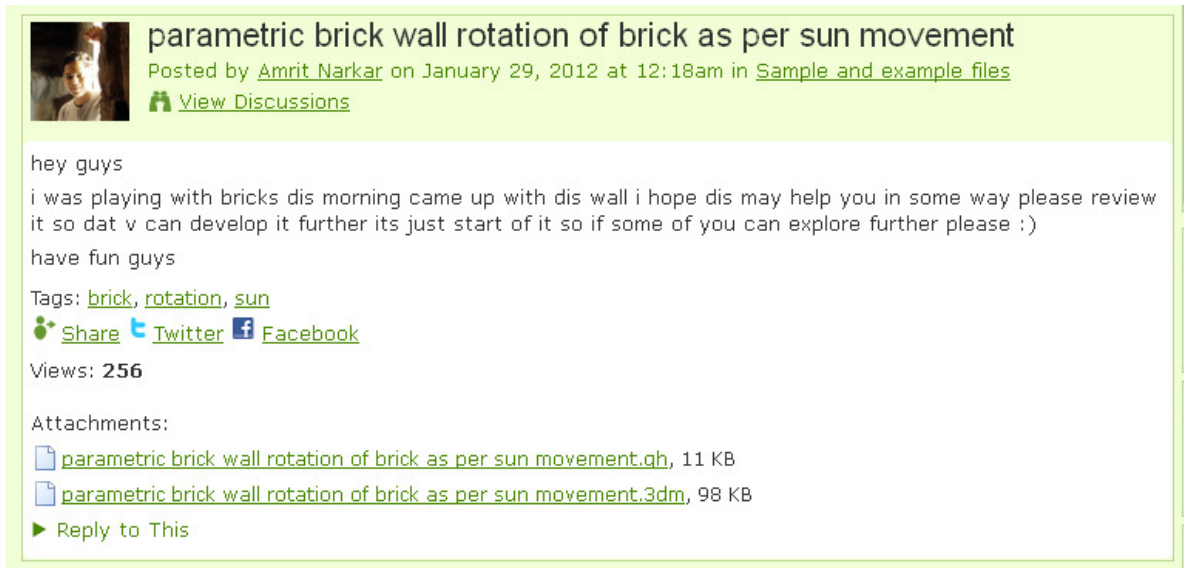


Figure 238 : capture d'écran d'un profil de membre sur [smartgeometry.com] (le 22-03-2012)



**parametric brick wall rotation of brick as per sun movement**  
Posted by [Amrit Narkar](#) on January 29, 2012 at 12:18am in [Sample and example files](#)  
[View Discussions](#)

hey guys  
i was playing with bricks dis morning came up with dis wall i hope dis may help you in some way please review it so dat v can develop it further its just start of it so if some of you can explore further please :)

have fun guys

Tags: [brick](#), [rotation](#), [sun](#)

[Share](#) [Twitter](#) [Facebook](#)

Views: **256**

Attachments:

- [parametric brick wall rotation of brick as per sun movement.gh](#), 11 KB
- [parametric brick wall rotation of brick as per sun movement.3dm](#), 98 KB

[Reply to This](#)

**Figure 239 : capture d'écran d'une conversation sur [grasshopper3d.com] (le 22-03-2012)**