



Urban Digital Twins Levels of Integration: From Conceptualization to Technical Implementation

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor
of Science (PhD) by

Imane Jeddoub

University of Liège

Faculty of Science, Doctoral College in Geography

April 2026

This dissertation is approved by:

Prof. Dr. Roland Billen

Supervisor

Prof. Dr. Rafika Hajji

Co-supervisor

Composition of the doctoral committee:

Prof. Dr. Roland Billen

University of Liège, Belgium

Prof. Dr. Rafika Hajji

IAV Hassan II, Morocco

Prof. Dr. Volker Coors

Hochschule Für Technik Stuttgart, Germany

Composition of the examination committee:

Prof. Dr. Christophe Debruyne

University of Liège, Belgium

Prof. Dr. Roland Billen

University of Liège, Belgium (Supervisor)

Prof. Dr. Rafika Hajji

IAV Hassan II, Morocco (Co-supervisor)

Prof. Dr. Volker Coors

Hochschule Für Technik Stuttgart, Germany

Prof. Dr. Hugo Ledoux

3D geoinformation research group, TU Delft, Netherlands

Prof. Dr. Jean-Paul KASPRZYK

University of Liège, Belgium

This research was conducted at the GeoScITY Lab of the Geomatics Unit, part of the University of Liège, from March 2022 to January 2026, and is funded by the Belgian National Funds for Scientific Research FNRS_2019_SIG3.0_PDR/OL T.0024.20.

© Imane Jeddoub, 2026 (ORCID 0000-0003-0632-1406)



ABSTRACT

Urban Digital Twins (UDTs) have emerged as a way to support data-driven urban planning, monitoring, and decision-making by enabling dynamic and integrated digital representations of cities and territories.

Yet, despite the growing interest in both academia and practice, the concept still lacks a clear definition and a shared technical basis. The absence of common standards, together with the diversity of implementation approaches, has led to different interpretations, fragmented practices, and difficulties in moving toward operational deployment. Among the challenges that still hinder the implementation of UDTs in practice, data integration is considered a major one.

UDTs rely on the integration of heterogeneous data coming from multiple sources, scales, and domains, including 3D city models, Building Information Models, Internet of Things data, and simulation outputs. However, current practices show that there is still no systematic or generic way to integrate these data. In many cases, this results in siloed implementations and limited interoperability. The challenge is amplified when considering the dynamic aspects of UDTs, which require continuous updates throughout their lifecycle, from creation to use and update.

This thesis examines how data integration is understood and implemented in UDTs. It first looks at the conceptual foundations of the field, highlighting the gap between theoretical definitions and what is actually developed in practice. It also shows the variety of integration mechanisms currently adopted. Based on this analysis, the thesis proposes a formalization of three levels of data integration: Level 1, the conceptual model level, where integration is achieved through schema extensions; Level 2, the database level, where data are adapted to fit the data model; and Level 3, the front-end level, where integration takes place at the visualization or client side. Together, these levels provide a framework to better understand and compare integration approaches across different UDT implementations.

Beyond conceptual contributions, the thesis develops and implements a technical framework to operationalize these integration levels. It proposes City2Twin, a lightweight and flexible UDT platform based on open standards such as CityJSON and web-based architectures. The platform is designed to support data integration, visualization, and analysis. It is tested through several use cases involving BIM models, IoT sensor data, and simulation outputs, making it possible to compare the advantages and limits of different integration strategies in terms of flexibility, interoperability, and maintainability.

The findings show that there is no single integration approach that fits all use cases. The most suitable level of integration depends on the application context, the nature of the data, and the requirements of the UDT lifecycle. By formalizing these levels and illustrating how they can be implemented in practice, this thesis contributes to a better

understanding of data integration within UDTs and provides guidance for the design of flexible and interoperable UDT systems.

Overall, this work positions data integration at the core of UDTs and provides a basis for future developments toward more mature and operational UDT ecosystems.

Keywords: Digital Twins, 3D city model, City information model, Spatial data infrastructure, Data integration, Interoperability, IoT, BIM, CityGML, CityJSON, 3DCityDB, CJDB, OGC SensorThings API, IFC, Simulation, Energy ADE.

RÉSUMÉ

Les jumeaux numériques urbains (JNU) ont émergé comme des dispositifs permettant de représenter, intégrer et explorer de manière dynamique les systèmes urbains, en combinant des données hétérogènes et des modèles à différentes échelles afin de soutenir la planification urbaine, le suivi et la prise de décision.

Toutefois, en dépit de l'intérêt croissant qu'ils suscitent tant dans le milieu universitaire que dans la pratique, ce concept demeure flou et ne s'appuie pas sur une base technique clairement établie. L'absence de normes communes, associée à la diversité des approches de mise en œuvre, a donné lieu à des interprétations divergentes, à des pratiques fragmentées et à des difficultés de déploiement opérationnel. Parmi les enjeux qui entravent encore la mise en œuvre des jumeaux numériques urbains dans la pratique, l'intégration des données est considérée comme l'un des plus importants.

Les jumeaux numériques urbains reposent sur l'intégration de données hétérogènes provenant de multiples sources, à différentes échelles et de différents domaines, notamment des modèles urbains en 3D, des modèles d'information du bâtiment, des données de l'Internet des objets et des résultats de simulations. Cependant, les pratiques actuelles montrent qu'il n'existe toujours pas de méthode systématique ou générique pour intégrer ces données. Dans de nombreux cas, cela se traduit par des implémentations en silos et une interopérabilité limitée. Le défi est amplifié lorsque l'on prend en compte les aspects dynamiques des jumeaux numériques urbains, qui nécessitent des mises à jour continues tout au long de leur cycle de vie, de la création à l'utilisation et à la mise à jour.

Cette thèse examine comment l'intégration des données est appréhendée et mise en œuvre dans les jumeaux numériques urbains. Elle se concentre tout d'abord sur les fondements conceptuels de ce domaine, en mettant en évidence l'écart entre les définitions théoriques et ce qui est réellement développé dans la pratique. Elle présente également la diversité des mécanismes d'intégration actuellement utilisés. Sur la base de cette analyse, la thèse propose une formalisation de trois niveaux d'intégration des données : le niveau 1, niveau du modèle conceptuel, où l'intégration est réalisée par le biais d'extensions de schéma ; le niveau 2, niveau de la base de données, où les données sont adaptées pour correspondre au modèle de données ; et le niveau 3, niveau de l'interface utilisateur, où l'intégration s'effectue au niveau de la visualisation ou du client. Ces niveaux fournissent ensemble un référentiel qui permet de mieux comprendre et comparer les approches d'intégration entre différentes implémentations des jumeaux numériques urbains.

Au-delà des contributions conceptuelles, cette thèse développe et met en œuvre un cadre technique permettant de rendre opérationnels ces niveaux d'intégration. Elle propose City2Twin, une plateforme jumeaux numériques urbains légère et flexible reposant sur des standards ouverts tels que CityJSON et des architectures web. Cette plateforme est conçue pour prendre en charge l'intégration, la visualisation et l'analyse des données. Elle est testée à travers plusieurs cas d'utilisation impliquant des modèles BIM, des données de capteurs IoT et des résultats de simulation, ce qui permet de comparer les avantages

et les limitations des différentes stratégies d'intégration en termes de flexibilité, d'interopérabilité et de maintenance.

Les résultats montrent qu'il n'existe pas d'approche d'intégration unique adaptée à tous les cas d'usage. Le niveau d'intégration le plus approprié dépend du contexte d'application, de la nature des données et des exigences du cycle de vie du jumeau numérique urbain. En formalisant ces niveaux et en illustrant comment ils peuvent être mis en œuvre dans la pratique, cette thèse contribue à une meilleure compréhension de l'intégration des données au sein des jumeaux numériques urbains et fournit des orientations pour la conception de systèmes de jumeaux numériques urbains flexibles et interopérables.

Dans l'ensemble, ce travail place l'intégration des données au cœur des jumeaux numériques urbains et fournit une base pour les développements futurs vers des écosystèmes jumeaux numériques urbains plus matures et opérationnels.

Mots clés : Jumeaux numériques, Modèle de ville 3D, Modèle d'information de la ville, Infrastructure de données spatiales, Intégration des données, Interopérabilité, Internet des objets, BIM, CityGML, CityJSON, 3DCityDB, CJDB, API OGC SensorThings, IFC, Simulation, ADE Énergie.

ACKNOWLEDGEMENTS

Throughout my academic journey, I have often been reminded that achieving fulfillment in life requires the right Element, Environment, and Energy. Looking back on my PhD years, I now realize that these were not abstract concepts, but lived experiences made possible largely thanks to the people who accompanied, supported, and guided me along the way.

My deepest and most sincere gratitude goes first of all to my supervisor, **Prof. Roland Billen**, who has been a true mentor and leader in every aspect. His scientific expertise, strategic vision, and high standards shaped both the direction and the quality of this work. Our numerous brainstorming sessions, his constant availability, and his rigorous yet constructive feedback were fundamental to the development of this thesis. For his trust, guidance, and unwavering support, both academic and personal, I am profoundly grateful.

I would also like to express my sincere appreciation to my co-supervisor, **Prof. Rafika Hajji**, who believed in me from the very beginning and recommended me for this PhD position. She consistently supported my academic choices and research direction with clarity, insight, and dedication. Her expertise, thoughtful advice, and encouragement were invaluable throughout this journey. Over time, our collaboration evolved into true teamwork, driven by shared ambition and tangible achievements. Her trust and mentorship played a decisive role in the successful completion of this research.

I am grateful to the members of my thesis committee for their time, expertise, and insightful feedback. In particular, I thank **Prof. Volker Coors**, whose expertise significantly strengthened my reflections on data integration and contributed to the scientific depth of this work.

I would also like to warmly thank the members of my thesis jury, **Prof. Christophe Debruyne** (University of Liège), **Prof. Hugo Ledoux** (TU Delft), and **Dr. Jean-Paul Kasprzyk** (University of Liège) for their time, their interest in my work, and their participation in the defense.

I extend my thanks to the University of Liège for providing the resources and academic environment that made this work possible. I gratefully acknowledge the **Fonds National de la Recherche Scientifique** (FNRS) for funding this research and providing the financial support that allowed me to fully dedicate myself to this PhD.

I am also grateful for the open content used in this thesis, including open-source software and open data. This work would not have been possible without the commitment of the open science community.

My sincere thanks go to my colleagues and friends, in no particular order. I would like to thank **Abderrazzaq, Zouhair**, and **Anass** for their valuable advice, collaborations, and countless discussions. Coming from similar backgrounds and sharing the same motivation, we aimed not only to advance research but also to represent Moroccan

engineers abroad with integrity and excellence. Our discussions were sometimes divergent, but always complementary, helping to refine our perspectives. I am deeply grateful for your presence, support, and, above all, your time. Beyond our academic collaboration, I am especially grateful that, over time, we also built sincere friendships.

My deepest gratitude goes to **Gilles-Antoine** for the valuable exchanges and feedback he provided throughout this thesis. I am also sincerely grateful to **Charline** for her support, kindness, and for making life as an expatriate in Belgium more welcoming. Her generosity made me feel at home. I also warmly thank **Thomas** for always being available to help whenever I needed it most. I am also grateful to my colleagues **Caneren**, **Nourhan**, and **Cadwal** for their collaboration and support.

I would like to thank the master's students I had the privilege to co-supervise, **Yassine Oudadda**, **Sara Tetou**, **Benirina Parfait Rafamantanantsoa**, and **Hiba Doi**. Their contributions were substantial, and I consider myself very fortunate to have worked with such talented and motivated young researchers.

I would also like to thank the researchers I met during my PhD, notably **Dr. Rushikesh Padsala**, **Dr. Binyu Lei**, and **Prof. Filip Biljecki**, whose work helped ensure that my developments remained aligned with the Urban Digital Twins state-of-the-art. I am also grateful to ISSeP and the thermodynamics laboratory (notably **Mazarine Roquet** and **Prof. Pierre Dewallef**) for our collaboration; their input was valuable in testing and validating the levels of integration explored in this thesis.

I am grateful to all past and present colleagues in the geomatics unit for the enriching exchanges throughout these years.

On a personal level, my deepest gratitude goes to my support system first and foremost, my **mother**. She was always there to encourage me, even during moments of doubt, reminding me that this journey was mine and that uncertainty is part of the process. Thank you for your unconditional support. I also thank my **father** for his encouragement and for believing in me. My heartfelt thanks go to my brothers, **Othmane** and **Yassine**, for the joyful moments we shared and for our shared ambition to make our parents proud.

I am deeply grateful to my dear friends **Majda (my caring friend)**, **Wafaa (my dedicated friend)**, **Hamza**, **Amal**, **Hafsa**, **Salah**, **Nabil**, **Zakaria**, **Taha**, and **Kamil** for always checking on me and for their constant encouragement. Finally, I extend my gratitude to both my families, **Jeddoub** and **Yahyaoui**, as well as to my uncles, aunts, and cousins, for their love and continuous support.

To all the people I met during this thesis, and to all those who contributed to my personal development, thank you!

Imane Jeddoub, 2026.

CONTENTS

ABSTRACT	3
RÉSUMÉ.....	5
ACKNOWLEDGEMENTS.....	7
CONTENTS	9
LIST OF FIGURES	13
LIST OF TABLES	15
LIST OF ABBREVIATIONS	16
CHAPTER 1: Introduction	19
Preamble	20
Context.....	21
Research questions	23
Thesis' scope.....	24
CHAPTER 2: Urban Digital Twins: State of the Art.....	27
Preface	28
2.1 Introduction.....	32
2.2 Background research and related work.....	34
2.2.1 Common definitions of DTs.....	34
2.2.2 Data integration challenges	35
2.3 Methodology.....	37
2.4 Digital Twins for Cities: Analysis of Definitions and Study of Concepts	38
2.4.1 Digital Twins for cities: Systematic literature review of definitions and concepts.....	39
2.4.2 A comparative literature review with related concepts: 3DCM, CIM, SDI	41
2.4.3 Digital Twins for cities: Survey analysis of concepts and key features.....	44
2.4.4 Summary and discussion of the DTs definitions and concepts.....	50
2.5 Digital Twin for cities: description, analysis and classification of current implementations.....	51
2.5.1 Digital Twins for cities initiatives: Study of realizations	51
2.5.2 Level of data integration between both worlds.	54
2.5.3 Data integration methods-based use cases.....	55
2.6 Discussion of findings	60
2.7 Urban Digital Twin Definition Adopted in This Thesis.....	62
2.8 Conclusion.....	63

CHAPTER 3: Data Integration across UDT Lifecycle: Conceptualization.....	65
Preface	66
3.1 Introduction.....	69
3.2 Data integration and related issues.....	71
3.3 Conceptualization of data integration levels for UDTs	73
3.3.1 Conceptual data model level (CDM)	73
3.3.2 Database level.....	74
3.3.3 Front-end level (client side)	74
3.4 Data integration levels: Pros and cons.....	76
3.5 Unpacking and illustrating data integration levels according to the DT lifecycle...	77
3.5.1 Creation of UDT	78
3.5.2 Use of UDT	82
3.5.3 Update of UDT	87
3.6 Data integration levels: discussion and findings.....	89
3.7 Conclusions.....	91
3.8 Subsequent work.....	92
CHAPTER 4: Rethinking UDT Architectures: A Full-Stack CityJSON-Based Platform	97
Preface	98
4.1 Introduction.....	101
4.2 Related work.....	101
4.3 Method.....	102
4.3.1 Study Area and Datasets.....	102
4.3.2 Configuration of the workspace	106
4.3.3 Data Preparation and Integration.....	106
4.3.4 External Data Integration	107
4.4 Data Storage and Management	108
4.4.1 Selection of the 3D Database	108
4.4.2 Importing static model to CJDB	110
4.4.3 Optimization of the CJDB Database.....	110
4.4.4 Integration of dynamic data.....	111
4.4.5 CityThings concept	112
4.5 Data Visualization	114
4.5.1 Choice of visualization tool and approach.....	114
4.5.2 3D Data Retrieval based on Giro3D framework.....	115
4.5.3 Dynamic Data Visualization	116
4.6 Various features integrated into the platform	117

4.6.1 Retrieving information related to an object by its ID	117
4.6.2 Retrieving information related to a specific surface	118
4.6.3 Multi-layer management.....	118
4.6.4 Attributes manipulation.....	119
4.6.5 Filtering objects.....	120
4.6.6 Optimization of visualization	121
4.6.7 User management.....	121
4.7 Results and Use Cases.....	121
4.7.1 City2Twin Platform.....	121
4.7.2 Use Cases	122
4.8 Discussion.....	124
4.8.1 Used Tools.....	124
4.8.2 Evaluation compared to Cesium-based approach.....	125
4.8.3 Evaluation of City2Twin flexibility and consistency	127
4.9 Conclusion.....	129
CHAPTER 5: Levels of Integration within UDTs: Implementations and Evaluation	130
Preface	131
5.1 Introduction.....	134
5.2 Background and Related work.....	136
5.2.1 Data Integration Overview	136
5.2.2 Data integration levels	138
5.3 Conceptualization and scenario definition.....	139
5.3.1 Scenario 1: Integrate an IFC model	139
5.3.2 Scenario 2: Integrate IoT Data into UDT	140
5.3.3 Scenario 3: Integrate simulation outputs into UDT	142
5.4 Use cases Implementation	144
5.4.1 Scenario 1: Integrate IFC model to UDT.....	144
5.4.2 Scenario 2: Integrate IoT data into UDT.....	148
5.4.3 Scenario 3: Integrate simulation results into UDT.....	153
5.5 Discussion, Practical Insights, and Recommendations.....	159
5.5.1 General discussion: Why do integration levels matter?.....	160
5.5.2 Lessons learned, practical insights and recommendations.....	161
5.6 Conclusion.....	162
CHAPTER 6: Conclusions and Future Perspectives	164
6.1 Findings and contributions.....	165
6.1.1 Research questions	165

6.1.2 Summary of the main contributions of the thesis	170
6.2 Research extensions.....	173
BIBLIOGRAPHY	192
CURRICULUM.....	204
LIST OF PUBLICATIONS.....	205

LIST OF FIGURES

Figure 1.1. Overview of the thesis structure.....	25
Figure 2.1. Labels of Digital Twin for cities in literature and in practice.	32
Figure 2.2. Steps and search queries to identify research articles about Digital Twin for cities in the urban and geospatial domain.....	37
Figure 2.3. Survey sections conducted to complement the scientific literature.....	38
Figure 2.4. Digital Twin concept: Maturity levels according to the level of data integration.....	41
Figure 2.5. DT and CIM approaches: (a) CIM and DT overlap the same conceptual and technical realities; (b) CIM is the input layer for implementing DT for cities; (c) DT is a specialization of CIM and a support technology for developing the conceptual framework.....	43
Figure 2.6. Participant's profile.....	46
Figure 2.7. The level of importance of DT characteristics according to the survey.	47
Figure 2.8. Differentiation of the levels of consent according to the participants' profiles about the concepts of 3DCM and DTs.....	48
Figure 2.9. The level of agreement regarding the statements related to CIM and DTs.....	49
Figure 2.10. The level of agreement regarding the statements related to SDI and DTs. ..	50
Figure 2.11. Levels of data integration: Level 1- Conceptual Data Model, Level 2- Database, Level 3- Application.	56
Figure 3.1. Conceptualization of Levels of Integration.	67
Figure 3.2. Levels of data integration: Possible ways for integrating data into UDT.....	73
Figure 3.3. The Urban Digital Twin lifecycle stages.	78
Figure 3.4. Current scenarios of data integration levels in the creation phase.....	79
Figure 3.5. Theoretical scenarios of data integration levels in the creation phase.	82
Figure 3.6. Current scenarios of data integration levels in the use phase based on an energy use case.....	84
Figure 3.7. Theoretical scenarios of data integration levels in the use phase.	86
Figure 3.8. Scenarios of data integration levels in the update phase.	87
Figure 3.9. Data integration levels for DT updates: (a) at the creation phase, (b) at the use phase.	88
Figure 3.10. Overview of the participants' profile and the UDT projects.....	93
Figure 3.11. Comparative analysis of UDT projects: spatial scale vs. lifecycle stage.	94
Figure 3.12. Comparative analysis of UDT projects: lifecycle stage vs. levels of data integration.....	94
Figure 3.13. Relevance of data integration levels (a) among practitioners and (b) for UDT implementation.....	95
Figure 4.1. City2Twin methodology overview.....	103
Figure 4.2. Retrieved API responses for sensor ID = 42: sensor metadata and location (left) and air quality observations (right).....	104
Figure 4.3. Distribution of air quality monitoring stations in the city of Liège (Source: OpenDataLiège).....	104
Figure 4.4. Air quality monitoring station in the OutreMeuse district (Study area).....	105
Figure 4.5. BelAQI hourly index classes based on the relationship between daily maximum hourly concentrations and daily average concentrations (maximum 8 hours for O3). (Source: irceline.be).....	105
Figure 4.6. UML diagram of the STA data model.....	108

Figure 4.7. Optimization of the CJDB database <i>ground geometry</i> .	111
Figure 4.8. Linking CJDB and STA Databases following the CityThings Concept.	113
Figure 4.9. The CityThings concept applied to a sensor in the study area: (a) Static attributes from the modeling process, (b) dynamic attributes from the STA DB.	113
Figure 4.10. UML sequence diagram of 3D Data Loading.	116
Figure 4.11. Example of retrieved data for a specific sensor.	116
Figure 4.12. Sequence Diagram for dynamic data analysis.	117
Figure 4.13. Retrieving information about a specific object (i.e., building).	118
Figure 4.14. Retrieving information about a specific surface (i.e., Roof Surface).	118
Figure 4.15. Managing multiple layers from various sources.	119
Figure 4.16. Attributes management: updating, adding, and deleting attributes.	120
Figure 4.17. a) Attribute filter: buildings of type «Building», trees < 20m, and roads for national use; b) Spatial filter: buildings located within 10m around a road segment.	121
Figure 4.18. The City2Twin overview: (a) layer management panel, (b) controls panel, (c) attribute and spatial filters, (d) urban analysis (air quality and energy simulation panel), (e) import external data, (f) documentation, (g) user management.	122
Figure 4.19. Air quality measurement stations classified based on the AQI value.	122
Figure 4.20. Time series analysis of air quality: (a) Concentrations of each pollutant at time t; (b) Temperature variations from 08/01 to 08/26; (c) Average hourly concentrations calculated between 08/19 – 08/26; (d) Concentration variations between 08/23 and 08/25; (e) Average daily concentrations calculated between 08/19 – 08/26; (f) Average hourly temperatures calculated between 08/20 – 08/25.	123
Figure 4.21. Visualization of integrated simulation data on the model.	124
Figure 4.22. Overview of the visualization architectures of the two approaches: (a) the CesiumJS-based approach and (b) the Giro3D framework-based approach.	126
Figure 5.1. Unpacking the integration levels across the UDT lifecycle.	139
Figure 5.2. Examples of BIM model integration levels across UDT phases.	140
Figure 5.3. Illustrative approaches of IoT feeds integration levels across UDT lifecycle.	142
Figure 5.4. Examples of integrating energy-simulation outputs across the UDT lifecycle.	143
Figure 5.5. The IFC models, (a) AC20-FZK-Haus and (b) AC20-Institute-Var-2.	144
Figure 5.6. Scenario 1 outputs: (a) Level 2-LoD 3.2 CityJSON model, (b) Level 3 -AC20-Institute-Var-2 IFC model.	148
Figure 5.7. The outputs of scenario 2: (a) Level 1- Lamppost with CityJSON v2.0 “Dynamizer” extension, (b) Visualization of “Dynamizer” in City2Twin, (c) Level 2- Integration air quality measurements in CityGML3.0, (d) Level 3- CityThings concept, a bidirectional link between the CityJSON model and the air quality sensors.	153
Figure 5.8. 3DCityDB workflow: Energy ADE 2.0 data integration and export.	155
Figure 5.9. The outputs of scenario 3: (a) Level 1- CityGML model with Energy ADEv2.0 visualized in Cesium-based platform, (b) Level 2- 3D CityJSON model using generics, (c) Level 3- OGC STA implementation, linking the CityJSON model with the energy STA service, (d) a lightweight front-end implementation following data mapping and linking using unique ids.	159
Figure 6.1. Decision-support framework for deriving semantic and geometric levels of integration.	171
Figure 6.2. Framework for selecting the recommended integration strategy based on semantic and geometric levels of integration.	172

LIST OF TABLES

Table 2.1. Description of categories to document the DT for city initiatives.....	52
Table 2.2. DT initiatives classification according to the data type and the maturity level of integration.....	54
Table 2.3. Data integration approach-based use cases and platforms.	58
Table 3.1. The pros and cons of data integration levels.	76
Table 3.2. Examples of data integration levels (Lv) in the creation phase using different data inputs.....	79
Table 3.3. Examples of data integration levels in the use phase based on an energy use case.....	84
Table 4.1. 3D city model of Outremeuse characteristics.....	103
Table 4.2. Comparison of database sizes and query times for CJDB and 3DCityDB.....	109
Table 4.3. Comparison of size and query time for 3800 buildings.....	110
Table 4.4. Comparison between CJDB database size and response time before and after rounding.....	110
Table 4.5. Creation of STA entities and their mapping to the case study measurement station.	111
Table 4.6. Comparative Analysis between JavaScript frameworks: CesiumJS, ThreeJS, and Giro3D.....	115
Table 4.7. Comparison between City2Twin and CesiumJS based approach.....	125
Table 5.1. BIM model overview.....	144
Table 5.2. Summary of dynamic air-quality datasets.	149
Table 5.3. Datasets overview, source and description.....	153
Table 5.4. Combinations of energy datasets and 3D city models and their implementations.....	154

LIST OF ABBREVIATIONS

Abbreviation	Full form
3DCityDB	3D City Database
3DCM	3D City Model
ADE	Application Domain Extension
AEC	Architecture, Engineering and Construction
API	Application Programming Interface
AQI	Air Quality Index
AR	Augmented Reality
BIM	Building Information Modelling
CDM	Conceptual Data Model
CDT	City Digital Twin
CFD	Computational Fluid Dynamics
CIM	City Information Modelling
CJDB	CityJSON Database
CRS	Coordinate Reference System
CZML	Cesium Markup Language
DB	Database
DM	Digital Model
DS	Digital Shadow
DT	Digital Twin
EPC	Energy Performance Certificate
EPSG	European Petroleum Survey Group (Coordinate System)
ETL	Extract, Transform, Load

EuroSDR	European Spatial Data Research
FME	Feature Manipulation Engine
FNRS	Fonds National de la Recherche Scientifique (Belgian research funder)
GenAI	Generative Artificial Intelligence
GeoSciTY	Geospatial Data Science and City Information Modelling
GIS	Geographic Information System
GML	Geography Markup Language
I3S	Indexed 3D Scene Layers
IFC	Industry Foundation Classes
IoT	Internet of Things
JSON	JavaScript Object Notation
LiDAR	Light Detection and Ranging
LoD	Level of Detail
LoI	Level of Integration
NMCA	National Mapping and Cadastral Agencies
OGC	Open Geospatial Consortium
SC	Smart Cities
SDI	Spatial Data Infrastructure
SQL	Structured Query Language
STA	SensorThings API
UIM	Urban Information Model
UDT	Urban Digital Twin
UML	Unified Modeling Language

"When you protect your purpose, your purpose protects you.
One of mine was the PhD."
Imane

CHAPTER 1: Introduction

Preamble

Research addressing complex urban systems must first provide the reader with a clear understanding of the context, motivations, and conceptual background necessary to follow the rest of the thesis. This chapter serves that purpose. It offers a broad contextualization of the thesis and the research project, targeting readers who may not have prior knowledge of smart cities, urban and city information modeling, and digital twins for cities. It does not aim to provide a comprehensive state of the art on urban digital twins (UDTs) and their related challenges that will be tackled within the scope of this thesis. Instead, it introduces key concepts, identifies challenges in current discourse and approaches related to digital twins for cities, and outlines the research questions that will be addressed in the following chapters.

This thesis draws inspiration from previous work by Dr. Gilles-Antoine Nys, defended in 2023 within the GeoScITY lab, which explored a paradigm shift in 3D city model management from traditional relational databases to document-oriented NoSQL solutions, with the aim of enabling a flexible web architecture that improves usability, interoperability, and maintenance. Continuing along this line, the present research takes a step forward by focusing on data integration as a key step toward Urban Digital Twins, investigating how heterogeneous urban data can be harmonized and integrated, and at which level such integration is most effective within a UDT. Furthermore, following the active contributions around CityJSON, we pursued the development of urban digital twins while maintaining flexibility as a guiding principle, mainly through lightweight, web-friendly data standards and architectures.

Funded by the F.R.S.-FNRS research project GIS3.0, this thesis addresses fundamental research questions related to how heterogeneous urban data and models can be represented, integrated, and accessed with the emerging concept of Digital Twins for urban and city environments. More broadly, this research aims to contribute, at least to some extent, to the evolution of geoinformation systems through a multidisciplinary and collaborative perspective on multisource data integration and interoperability.

Within the framework of the thesis, several research contributions and standardization initiatives have supported conceptual reflections and technical choices for implementations. On the conceptual side, the work of (F. Biljecki, 2017) on levels of detail in 3D city models provides a foundation for understanding the multi-scale nature of urban representations and their implications for modeling, analysis, and for use cases and urban applications. In parallel, research on multisource data integration conducted by (Noardo, 2022) has helped frame the integration challenges that were identified in the framework of BIM-GIS integration providing insights directly relevant for urban digital twins, which inherently require the integration of heterogeneous urban datasets. This work provides a strong conceptual framework for addressing one of the main challenges of UDT, namely data integration and interoperability, and proposes guidelines for technical workflows that support automation and practical implementation.

On the technical side, the work of the 3D Geoinformation Research Group at TU Delft, namely their contributions around CityJSON and its supporting open ecosystem of tools,

has been valuable for this thesis, particularly in chapter 4 and chapter 5, where they support the development of the City2Twin platform and related components, while keeping flexibility and reusability as guiding principles.

This introduction aims to equip the reader with the conceptual elements necessary to understand the scope, motivations, and structure of the thesis, while setting the ground for more technical developments that follow.

Context

Over the past decade, the smart city paradigm has emphasized the need to use dynamic data to support urban planning, efficient operations, and inclusive decision-making. Many smart city initiatives have been launched, often centered around open data portals, sensor networks, or urban dashboards. While these approaches demonstrated their potential for fostering data-driven city infrastructures, the concept of smart city is often grounded in organizational and political considerations rather than a clearly defined technical implementation. As a result, the smart city vision lacks a well-defined technical foundation capable of supporting complex analyses, cross-domain integration and long-term decision-making. In this context, the concepts of City Information Modelling (CIM) and Urban Digital Twins (UDTs) have gained significant interest in both geospatial and urban domains. Both aim to translate the smart city vision into operational solutions by providing digital representations of cities while enabling analysis, simulation and cross domain decision-making. Chronologically, 3D city models and CIM emerged earlier within the geospatial community. The concept of UDTs gained momentum later, particularly from around 2016 onward. Since then, UDTs have attracted growing interest across a wide range of disciplines, often serving as an umbrella concept and catch-all term that brings together ideas from Geographical Information Systems (GIS), Building Information Model (BIM), Smart Cities (SCs), and Spatial Data Infrastructures (SDIs). At the same time, a shared and standardized definition of Urban Digital Twins has not yet fully matured, and the concept remains under active discussion. Many studies published since 2022 have attempted to clarify the concept of UDTs, to move toward a common understanding, and to articulate their similarities and divergences with related concepts. This gap is addressed in this thesis and is discussed in detail in chapter 2. Nevertheless, to facilitate understanding at this stage, a preliminary definition is provided below to give an initial idea of what an urban digital twin represents.

Digital twin is “a digital representation of a particular physical entity or a process with data connections that (1) enables convergence between the physical and digital states at an appropriate rate of synchronization, (2) has the capabilities of connection, integration, analysis, simulation, visualization and optimization, and (3) provides an integrated view throughout the life cycle of the physical entity or the process” [Source: ISO/IEC WD 30173]

Extending this definition to the urban and geospatial fields, UDT can be broadly defined as a virtual representation of the city and its subsystems, continuously fed by real-time and near real-time data coupled with advanced simulation modeling allowing the support of monitoring and prediction for a variety of stakeholders and use cases.

The emergence of UDTs reflects the growing need for more expressive, functional and integrated representations of cities. Historically, urban representations relied on two-dimensional maps or three-dimensional geometric models, primarily designed for visualization and planning. While valuable, these representations were often static snapshots of reality and limited in their analytical or operational capabilities. In contrast, semantic 3D city models show their potential to advance analytical, predictive, or operational urban applications (Biljecki et al., 2015). By combining geometry with explicit semantics, relationships, and levels of detail, they provide a spatio-semantically coherent representation of the urban environment that supports advanced analysis and modelling. Their increasing availability has reinforced their role as a foundational component for more integrated and dynamic urban representations (Wysocki et al., 2024). It is within this context that urban digital twins gain traction, building on 3D city models to address issues of data silos, fragmented workflows, and limited interoperability, while moving toward more integrated and dynamic representations of cities. The Urban Digital Twin concept has gained broader appeal due to its intuitive metaphor and its promise of unifying previously disconnected data, models, and processes by mimicking city infrastructures and dynamics. However, this wider adoption also highlights persistent challenges, particularly for geospatial data providers and urban modelers, related to data access, interoperability, and integration.

Since its introduction to the geospatial domain, the term “digital twin” has been used in a variety of ways, ranging from simple 3D city models to complex platforms integrating real-time sensor data, simulations, and decision-support tools. Despite the lack of a universally accepted definition, there is general agreement that an urban digital twin relies on standardized semantic 3D city models, including urban objects with explicit geometric and semantic information, and supports the integration of dynamic data and analytical processes. In this sense, digital twins build directly upon existing geospatial modelling practices rather than replacing them. As a result, many challenges in semantic 3D city modeling such as data completeness, geometric validity, semantic consistency, interoperability across standards and software, as well as the maintenance of the data infrastructure, remain central to data integration, as they directly shape the integration approach and its effectiveness.

Building on this natural evolution of existing geospatial models and infrastructures towards integrated, dynamic and connected digital representations brings data issues to the foreground. To be operational, a UDT needs to integrate and handle heterogeneous datasets derived from many sources and spanning different spatial and temporal scales. The UDT is mainly a data-driven and dynamic infrastructure that needs to harmonize them into a common representation of the city, and it must expose them in forms that are usable by many applications and decision processes. As with previous concepts in the geospatial domain, these challenges must be addressed through appropriate modelling principles, tools, and standards capable of supporting real-world applications. Despite their rapid proliferation, data integration remains one of the major challenges facing UDTs. Integrating heterogeneous, multi-source, and dynamic datasets into a coherent and interoperable representation continues to limit the reliability and usability of UDTs in practice.

Addressing this gap requires analyzing current data integration practices within UDTs, identifying their associated challenges, and investigating how a UDT can be designed to

integrate heterogeneous data while remaining flexible enough to support diverse urban applications and facilitate data access and maintenance. This is precisely where this thesis begins. It is based on both conceptual and technical reflections to produce results that are meaningful both for research and practice.

Research questions

The emergence of Urban Digital Twins and their growing relevance for cities and territories worldwide significantly increase data integration challenges. Although data integration is widely recognized as the main technical challenge in practice, comprehensive research that studies data integration approaches in detail within the UDT context remains limited. In particular, the UDT lifecycle is rarely taken into account when analyzing integration mechanisms. As cities move toward federated systems and increasingly adopt a system-of-systems perspective, there is a growing interest in understanding how data integration is performed and at which level it occurs. Formalizing integration approaches is recommended to avoid unnecessary acquisition of new datasets, promote the reuse of existing data, and generate value through integration and enrichment. The aim of UDTs is not to further increase data acquisition costs, but to break data silos and models and to move toward a coherent digital representation of the city through shared semantics and vocabularies.

Current implementations reveal the lack of a generic and systematic approach to data integration. On the one hand, data integration is carried out through a wide range of approaches, as will be discussed in this thesis. On the other hand, the data to be processed varies from one use case to another, leading to different integration mechanisms. Moreover, similar integration challenges have been encountered for more than a decade in related concepts such as semantic 3D city models, spatial data infrastructures, and City Information Modelling, yet achieving seamless integration and improved interoperability remains difficult in practice. This highlights the need to address data integration in a comprehensive manner, particularly when semantic 3D city models are used as the integration basis for UDTs.

Standardized models such as CityGML show potential for supporting data integration through Application Domain Extensions (ADEs), depending on the use case. However, as Urban Digital Twins evolve toward more dynamic and distributed systems, additional approaches to data integration are increasingly being considered. In this context, web-based technologies, semantic web standards, and Open Geospatial Consortium standards are often regarded as complementary mechanisms for enabling interoperability and data exchange. Nevertheless, their practical use within UDTs remains heterogeneous, and their role in supporting data integration across the UDT lifecycle is not yet clearly established.

The lack of research on this field opens the way to a central question: **How can urban digital twins be designed to integrate heterogeneous data while remaining flexible enough to support diverse urban applications and facilitate data access and maintenance?**

The main research question is further divided into a set of specific questions addressing both conceptual and technical aspects of data integration within UDTs.

Conceptual research questions

The conceptual part of research aims to address the following questions:

RQ1. How should a UDT be defined with respect to data integration and interoperability requirements, and to what extent does real-time interaction shape this definition?

RQ2. How is data integration addressed within the emerging concept of UDTs? Does the UDT paradigm introduce new data integration challenges compared to previous geospatial and urban modeling approaches?

RQ3. How can data integration be defined and formalized according to different approaches across the UDT lifecycle (creation, use, and update)?

Technical aspects and research questions

On the technical side, this research investigates the following aspects:

RQ4. How can an Urban Digital Twin framework be developed to support different levels of data integration in practice?

RQ5. How can different levels of integration be implemented across various UDT use cases?

RQ6. How can flexible and lightweight UDT architecture support, illustrate and operationalize different data integration levels?

Addressing these questions is particularly challenging, as Urban Digital Twins need to balance, on the one hand, rich semantic representations based on standardized urban data models and, on the other hand, the need for flexible and web-friendly solutions that can be deployed, maintained, and reused across different use cases and applications. Each of these questions is addressed in the following chapters.

Thesis' scope

This thesis is structured around four main chapters as illustrated in **Figure 1.1**, which aim to answer the research questions presented earlier, thereby contributing to the field of data integration and interoperability within Urban Digital Twins.

Besides the introduction (chapter 1) and the conclusion and future perspectives (chapter 6), the thesis is structured into two main parts, each consisting of two chapters. These chapters are based on three journal articles and two conference papers produced in this thesis, adapted and integrated into this manuscript. Each chapter is introduced with a preamble that summarizes the work and presents its context, aims, and main findings. In addition, each chapter defines its scope and related work and possible extensions for future work. Some parts have been updated since their original publication; for example, the literature review in chapter 2 has been extended to reflect recent research studies.

Where relevant, subsequent work building on the main articles such as additional journal or conference papers, related developments, and projects in which we were involved is also described at the end of each chapter. These complementary contributions strengthen the overall research process and help illustrate the relevance of the levels of integration within UDT both from practitioners and academia.

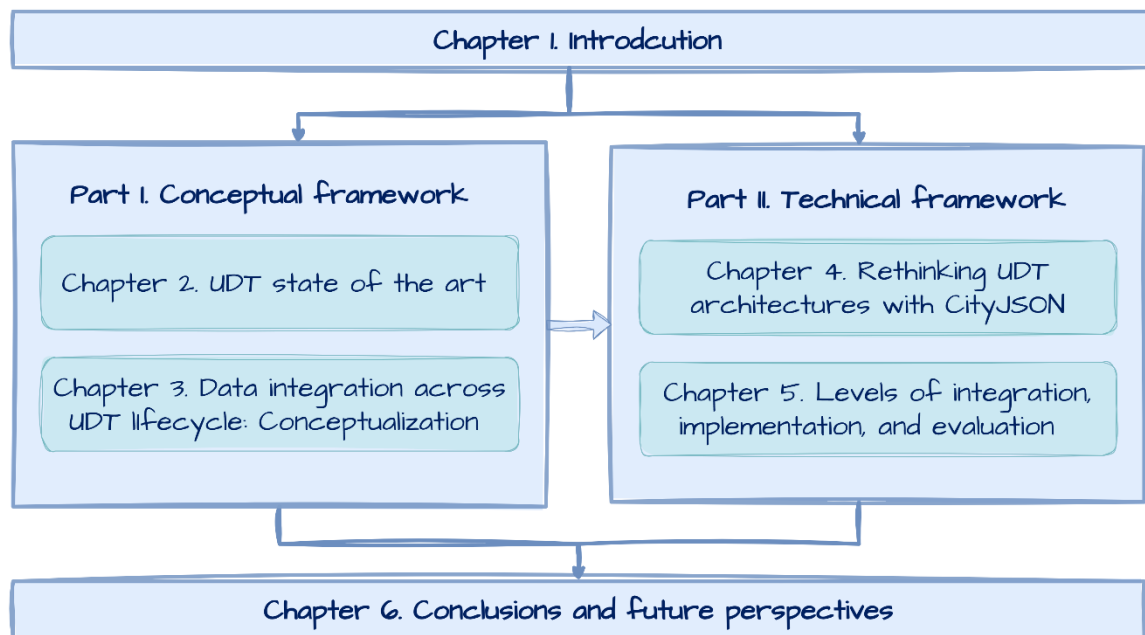


Figure 1.1. Overview of the thesis structure.

CHAPTER 2: reviews the state of the art on Urban Digital Twins, focusing on the lack of a common definition, the discrepancies with related concepts (e.g., 3D city models, City Information Modelling, and Spatial Data Infrastructures), and the gap between conceptual definitions and current implementations. Based on a literature review and an online survey, it analyses current UDT initiatives from a technical perspective, with a specific focus on data integration, and shows that current practices remain heterogeneous and based use cases. The chapter concludes by highlighting data integration as a central challenge across the UDT lifecycle and by providing an analytical basis for the following chapters.

CHAPTER 3: introduces the Levels of Integration (LoI) and discusses different data integration approaches in UDTs by framing integration as an nD process that needs to consider data heterogeneity across dimensions (i.e., 2D/3D geometry, time, and semantics). Building on this background from existing UDT frameworks and implementations, the chapter proposes three integration levels (e.g., conceptual data model, database and front-end levels) and relates them to the main stages of the UDT lifecycle (creation, use, and update). Through a set of current and theoretical integration scenarios, it analyses the challenges and potentialities of each level of integration and provides conceptual guidelines to support UDT designers in planning integration strategies. The chapter also highlights that current UDT implementations mainly focus on the creation and use phases, while integration for long-term update and maintenance remains uncovered. This limitation motivated us to launch a [survey](#) to assess the relevance and applicability of these integration levels for both practitioners and academia, and to better understand how the geoinformation community approaches their implementation and update over time. This follow-up work is presented as an extended contribution at the end of Chapter 3. This chapter concludes the conceptual part of the thesis by addressing and clarifying the conceptual research questions formulated in the research question section.

CHAPTER 4: addresses the operational implementation of Urban Digital Twins by proposing a shift from common CityGML-based architectures, typically relying on 3DCityDB, toward lightweight and web-oriented solutions. The chapter presents the City2Twin platform as an open-source Urban Digital Twin framework designed to support data integration, storage, visualization, and analysis using open-source tools. By adopting **CityJSON** as the core encoding for 3D city models and evaluating the **CityJSON Database (CJDB)** as an alternative backend, the chapter demonstrates how lightweight data representations and direct database-client interaction can simplify workflows, improve maintainability, and better support dynamic urban applications. Beyond providing a standalone implementation, the **City2Twin** platform establishes a technical foundation for subsequent developments, preparing the ground for the implementation and evaluation of data integration levels presented in Chapter 5. Through comparative experiments and design choices, the chapter highlights the limitations of existing approaches and identifies key technical requirements for developing flexible, scalable, and interoperable Urban Digital Twin platforms.

CHAPTER 5: implements the proposed Levels of Integration (LoI) framework by applying different data integration approaches within an Urban Digital Twin across its lifecycle and discusses the main outcomes. Building on the conceptual framework introduced in Chapter 3 and the City2Twin platform developed in chapter 4, the chapter analyses how heterogeneous data are integrated in practice. Through concrete and reproducible integration scenarios covering BIM model, dynamic IoT sensor data, and simulation outputs, it analyzes the technical choices, trade-offs, and implications of each integration level in terms of semantics, flexibility, maintainability, and interoperability. The chapter derives practical insights and guidelines to support integration approaches and to better align them with operational UDT requirements and evolving, federated UDT architectures.

CHAPTER 6: concludes the thesis by summarizing the key findings and contributions of the research and by addressing the research questions introduced earlier. It discusses how the conceptual framework and technical implementations presented in the previous chapters contribute to the current state of the art on data integration and interoperability within UDTs. The chapter outlines the limitations of the proposed approaches and presents directions for future work, particularly in relation to the evolution of UDTs, integration approaches across the lifecycle, and the transition toward scalable and federated UDT ecosystems.

Note on contributions

This work was mainly carried out by me as first author, under the supervision of my promoters. Parts of the work were also conducted within the framework of master's theses that I co-supervised together with the thesis supervisors, and which contributed to the development of the technical framework presented in this thesis. In particular, the work of *Rafamatanantsoa Benirina Parfait* and *Yassine Oudadda* contributed to the development and implementation of components related to the City2Twin platform and associated conference publications. In this manuscript, the pronoun "we" is used to reflect this collaborative research context, while acknowledging my primary contribution. All related publications and contributions are cited accordingly.

CHAPTER 2: Urban Digital Twins: State of the Art

Preface

This chapter is rooted in a key observation concerning the gap between the theoretical definitions of Digital Twins for cities and their implementation in practice. Clarifying concepts is a good starting point to ensure that the underlying concepts are framed consistently, in order to reduce ambiguity and establish realistic expectations around Digital Twins for cities, while supporting implementations that move beyond pilot projects and proof-of-concept demonstrations. When this research was initiated in March 2022, the concept of Digital Twins for cities was still emerging and was often used interchangeably with related urban and geospatial concepts. Both the scientific literature and practical implementations revealed significant divergence, conceptual overlaps, and a clear gap between theoretical definitions and real-world applications. At that time, research on Digital Twins for cities was still at an early stage, which explains the limited number of related studies presented in the scientific literature from the geoinformation domain, and the fact that only a few studies attempted to critically examine how these concepts were being implemented in practice.

The aim of this chapter is to answer a part of the conceptual research questions (RQ), namely,

RQ1. How should a UDT be defined with respect to data integration and interoperability requirements, and to what extent does real-time interaction shape this definition?

Building on these research questions, the chapter brings together and reflects on two complementary perspectives: (1) what researchers have proposed through definitions and conceptual frameworks, and (2) what practitioners are developing through real initiatives and technical deployments. Through a systematic review of scientific and grey literature, supported by an exploratory international survey, this chapter synthesizes how Digital Twins for cities are currently defined, how their key features are described, and where the main points of divergence emerge. In particular, it examines the overlaps and discrepancies between Digital Twins and related approaches such as 3D City Models, City Information Models, and Spatial Data Infrastructures, which are concepts that often appear alongside Digital Twins and are sometimes unintentionally used as substitutes.

A key reflection on data integration is introduced in this work as a foundation to initiate the investigation of data integration within Urban Digital Twins. Data integration is considered not only as a technical concern, but also as a specific evaluation criterion for determining whether an initiative truly moves from 3D City Models toward the Digital Twin ambition, namely, mutual interaction between the physical city and its digital counterpart. By reviewing and comparing documented initiatives, the chapter highlights how current implementations tend to adopt different integration approaches and maturity levels, and it proposes a conceptual framework to classify these initiatives according to how and where data integration is performed.

This chapter is based on the article *“Digital Twins for Cities: Analyzing the Gap Between Concepts and Current Implementations with a Specific Focus on Data Integration”* (Jeddoub et al., 2023). The article aimed to (1) review definitions of Digital Twins for cities and refine the terminology by extracting key features and comparing the concept with related approaches, namely 3D City Models (3DCM), City Information Models (CIM), and Spatial Data Infrastructures (SDI); (2) analyze the gap between theoretical conceptualizations

and what has been implemented in practice by assessing existing initiatives according to nine criteria (scale, data sources, city modelling, level of detail, purposes, visualization platforms, simulation capabilities, update frequency and methods, and implementation status); and (3) introduce an approach to classify Digital Twin initiatives according to both their maturity and their level of data integration.

This chapter presents the studies conducted on Urban Digital twins up to the date of publication of the original paper in 2023. Since then, additional research has been conducted, some of which builds upon the findings of this work. In this preamble, it is therefore necessary to give an overview of the recent contributions that further refine the definition and the conceptual setting of the UDT. This update aims to ensure that the chapter remains relevant and reflects the current state of the art.

For instance, (Abdelrahman et al., 2025) provides a comprehensive and well-structured analysis of the concept of digital twins (DTs) within the built environment, addressing its terminological challenges strengthened by the lack of consensus in definitions. The study offers novel and systematic methods to clarify its definitions, including Natural Language Processing (NLP) techniques and systematic reviews, to analyze definitions across disciplines and over time. They distinguish between Long-Term Decision Support (LTDS) digital twins and High-Performance Real-Time (HPRT) digital twins, thereby shifting the discussion from strict definitions toward functional categorization. Similarly, (Haraguchi et al., 2024) introduced a maturity model to assess the technological and governance advancement of UDTs, emphasizing that UDTs exist along a spectrum of data lifecycle stages. Recent work also argues that debates around definition and typology are still a challenge and proposes *twinning* as a lens to examine how UDT initiatives reshape urban representation and governance over time (Macq et al., 2025). Moreover, (Raes and McAleer, 2025) deliver a book on open and interoperable Local Digital Twins (LDTs), where a definition of the concept LDTs is proposed.

In addition to scholarly contributions, NMCA initiative is launched to provide a Digital Twin cookbook that primarily aims to consolidate and operationalize the concept of UDTs. The Digital Twin Cookbook for NMCAs¹ developed within the EuroSDR framework primarily seeks to establish a shared understanding and definition of UDTs. It also discusses requirements, stakeholder roles, and use-case development to support implementation in practice. Furthermore, standardization organizations such as IEC/CD TS 63526 are working towards defining concepts and setting the technical specifications. In this recent context, these contributions suggest a gradual shift in literature. Rather than attempting to establish a single, universal definition of Urban Digital Twins, scholars increasingly propose typologies and assessment frameworks that aim to highlight the discrepancies in integration, applications, governance structure, and operational scope. At the same time, these studies also emphasize that reaching a shared definition remains challenging. Urban Digital Twins tend to introduce additional layers of technical and organizational complexity, bringing together stakeholders with distinct interpretations, expectations, and needs. Yet, despite this lack of consensus, UDT initiatives continue to proliferate and evolve in practice. In this context, establishing agreement on a set of minimum key components as a baseline for identifying what can be called an Urban Digital Twin already represents a significant and constructive step. From this perspective, the

¹ <https://github.com/jackmetcalfe1/digital-twin-nmca-cookbook>

classification approach proposed in this chapter, particularly the emphasis on data integration levels and implementation maturity, remains consistent with and supported by subsequent contributions.

Beyond the academic scope of the original work, the chapter is also positioned within the broader thesis narrative by connecting its findings to the subsequent chapters. By the end of this chapter, the reader will have a clearer understanding of how Digital Twins for cities are framed in the literature, what gaps and limitations characterize current practice, and what long-term vision is implied by the concept of a UDT. The chapter also provides a reference point for interpreting and positioning Urban Digital Twin initiatives, offering conceptual grounding and a structured way to evaluate what is being built today against what is promised in theory. Supplementary materials are provided in APPENDIX A.

Based on the article (Jeddoub et al., 2023)

Digital Twins for Cities: Analyzing the Gap between Concepts and Current Implementations with a Specific Focus on Data Integration

Jeddoub, I., Nys, G.-A., Hajji, R., & Billen, R. *International Journal of Applied Earth Observation and Geoinformation*, 122, 103440. doi: 10.1016/j.jag.2023.103440 (2023).

Abstract: Digital Twins (DTs) for cities represent a new trend for urban and geospatial fields. Currently, some DTs implementations are taking place. Nevertheless, the whole concept remains ambiguous and presents some differences that need to be addressed. The aim of this article is to bridge the gap between DTs definitions and current implementations. This review was conducted through a scientific literature review and an online survey. The study collects DTs for cities definitions and compares them with related concepts used jointly in the literature. It puts them together through an in-depth analysis since they express similarities and various discrepancies. As our study highlights the most documented DTs initiatives for cities according to 9 comprehensive categories, a new approach assessing the initiatives is proposed to evaluate the data integration methods used in the current realizations. Three levels are suggested: a conceptual schema model-based level (the data are integrated into the top level of the DT, i.e., extending the schema model to cover new features or themes); a database-based level (data are integrated in order to feed or update specific attributes or classes); and an application-based level (the data are integrated into the application generally at the viewer level).

Keywords: Digital Twins; 3D city model; City information model; Spatial data infrastructure; Data integration.

2.1 Introduction

Digital Twins (DTs) rely unanimously on three major components: a physical (entity, system, process), its digital representation and seamless data connections that bind the digital and real counterparts together (Grieves, 2016; VanDerHorn and Mahadevan, 2021). While the concept of a DT is quite an old term in several industries (Ammar et al., 2022; Mylonas et al., 2021; Tao et al., 2019; Xiong and Wang, 2022), it is starting to gain a significant interest in the urban and geospatial context from 2016 and onward (Ketzler et al., 2020; Stoter et al., 2021). Their evolution in the urban setting results in a continually increasing variety of terms (see **Figure 2.1**) and different fit-for-purpose definitions. Hence, the original concept of DTs has been adjusted (P. Alva et al., 2022) to meet several urban requirements. This initiative led to an ineffective implementation of the concept and a huge ambition of the added value of DTs for cities that could be unachievable.

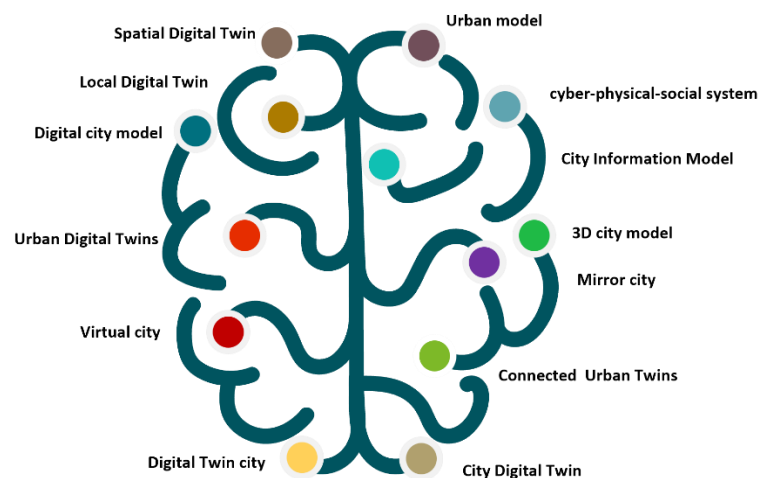


Figure 2.1. Labels of Digital Twin for cities in literature and in practice.

DTs for cities come from the willingness of the Smart Cities (SCs) initiatives to introduce digital technologies, to implicate various players (governments, private parties, citizens) and to generate insights for better decision-making through a range set of simulations and urban analysis. Furthermore, due to advances in technology, availability of spatial and non-spatial data and the global evolution of the virtual simulation technologies, more and more complex physical and dynamic systems, including cities, could be handled using DTs technologies (Grieves, 2022; Ketzler et al., 2020; Tomko and Winter, 2019). Nevertheless, the process of replicating this concept from other industries in the urban environment is not straightforward due to the complexity, the spatial and temporal urban scale. Narrowing DTs for cities scope to the geospatial domain and the Architecture Engineering Construction (AEC) field, such cities complexities were already approached 20 years ago, namely with 3D City Models (3DCM) (Billen et al., 2014), Building Information Models (BIM) (Hagedorn, 2007), Virtual City Models (VCM) (Batty and Hudson-Smith, 2006) and recently City Information Models (CIM) (Omrany et al., 2022). This leads us to the following questions: **What is exactly a DT for cities?**

There is an active debate regarding DTs for SCs, since DTs face challenges as revealed in the review by (Biljecki et al., 2015), including **no consensus on the exact definition**

(Saeed et al., 2022; Shahzad et al., 2022), **vagueness on the characteristics** (Sepasgozar, 2021), **terminological ambiguity** (Ketzler et al., 2020), **different forms and outputs** (Ferré-Bigorra et al., 2022), **plenty of technical approaches** that are domain specific (Lehtola et al., 2022) and **various technical and non-technical challenges** that hinder its full deployment (Lei et al., 2023). Moreover, there is still a gap related to a complete model and data interaction between the two counterparts.

Another challenge that hinders the full deployment of DT for cities is the lack of a generic data integration framework (Shahat et al., 2021). It is important to highlight the motivation towards the implementation of the DT for cities, which is breaking the silo-based approach to integrate the data and the models in an open, harmonized, interoperable ecosystem (Bauer et al., 2021; Nochta et al., 2021, 2019; Petrova-Antonova and Ilieva, 2021). Hence, data integration is one of the relevant themes that can be proposed to evaluate the potentials of DTs for cities (Botín-Sanabria et al., 2022; Papyshv and Yarime, 2021; Shahat et al., 2021; Xia et al., 2022). However, up to date an in-depth review of how the data is integrated in DTs for cities has not been discussed from a geospatial perspective.

The contributions of this chapter are as follows:

- (1) Reviewing DTs for cities definitions and refining the terminology by extracting the key features and comparing the DT for cities with other concepts, namely 3D City Models (3DCM), City Information Model (CIM) and Spatial Data Infrastructure (SDI).
- (2) Analyzing the gap between the theoretical findings of concepts and what has been implemented in practice. It is done by assessing current DTs initiatives in general and according to 9 criteria (scale, data sources, city modelling, level of detail, purposes, visualization platforms, simulations experiences, frequency and update methods, and status).
- (3) Introducing a new approach to classify DT for cities initiatives according to their level of maturity and their level of data integration.

To reach the goals of this work, our study followed three main steps: (1) a systematic literature review on DTs for cities conducted in scientific databases i.e., Scopus, (2) a review of grey literature (e.g., reports, and DT initiatives web pages) and (3) an exploratory online survey. Setting the survey is motivated by some differences in the current deployment of the DT concept noticed between different DT initiatives. Furthermore, the research regarding DTs for cities is at an early stage, which explains the small, related works presented in the scientific literature. Hence, the survey is thus of great interest to bridge some information gaps in relation to certain initiatives that are poorly documented.

This chapter is structured as follows. Section 2.2 briefly introduces the background of DT concept for cities in general and from a data integration perspective. Section 2.3 explains the research method conducted in this article, including a systematic review, an overview of grey literature and an online exploratory survey. Section 2.4 studies the definitions of DTs for cities according to the literature and analyses the DTs for cities concept with other related concepts 3DCM, CIM and SDI. A survey analysis is conducted as well to provide the findings of the questionnaire. Section 2.5 compares the current DTs for cities initiatives from a technical standpoint according to various criteria and classifies the current implementations according to their maturity level and data integration level proposed in

the scope of this work. Section 2.6 highlights the findings and discusses the results. Section 2.7 presents the UDT definition used in this thesis. Section 2.8 concludes this work and gives an outlook for future studies.

2.2 Background research and related work

While DT has many challenges, we focus only on two major aspects: **definition and data integration**. This enables us to discuss the differences between the theoretical concept and what is being developed in practice. For this purpose, we split the related work into two sections. Section 2.2.1 presents the concept of DT and explains challenges related to its definition. While Section 2.2.2 focuses on data integration and explains how the data has been historically embedded in a virtual city replica.

2.2.1 Common definitions of DTs

The concept of DT for cities is generally perceived as a main driver of digital transition across various disciplines, including urban and geospatial fields. To facilitate consensus and avoid the mishandling of the concept of DTs for cities across various stakeholders and domains, **a refinement of the definitions is a good starting point**. (VanDerHorn and Mahadevan, 2021) attempted to develop a unified definition and characteristics of DTs. After reviewing **46 DTs definitions** derived from the literature, the authors adopted broadly the following definition: a *"Digital Twin is a virtual representation of a physical system (and its associated environment and processes) that is updated through the exchange of information between the physical and virtual systems."* The definition can be projected into any specific domain. So, **how about defining DTs for cities?**

Recently, some studies on DT for cities were conducted to help overcome the challenges that DT for cities is tackling. In their work, (Ellul et al., 2022), motivated by the growing interest in the topic of DTs and its relevance to the European Spatial Data Research network (EuroSDR²) and geospatial communities, workshop sessions³ were conducted followed by a questionnaire⁴ to better understand how DT is defined and used in practice among different stakeholders and to provide insights to the National Mapping and Cadastral Agencies (NMCA). A considerable number of participants share their thoughts regarding the challenges and opportunities offered by DTs from NMCAs and non-NMCAs. One of the main goals is to analyze the responses that define DT for cities and list the predominant components. Hence, most of the responders agreed that DT is *"a realistic digital representation of physical assets, processes and systems."* (Lei et al., 2023) present another larger related work that addresses DT challenges for cities. The authors identified a list of challenges and classified them into 14 technical and 9 non-technical perspectives derived from academia and practice. A mapping of the challenges is presented according to the DT's lifecycle as defined in their work. Their findings were based on a multidimensional method: a systematic literature review and a Delphi survey across experts. Thus, creating DTs for cities is much more than just creating a virtual replica of city objects. It aims to construct a digital, living, interconnected ecosystem that interfaces with the real-world through continuous enrichment between the physical world and its digital replica. In contrast, with the knowledge gained from the SCs initiatives, often

² <http://www.eurosdrr.net/about>

³ <http://www.eurosdrr.net/workshops/digital-twins-nmcas>

⁴ <http://www.eurosdrr.net/workshops/follow-workshop-digital-twins>

focused on static 3D modeling and the increasing number of DT implementations in other industries, the construction of city DT requires **a data foundation and a technical framework** (Deren et al., 2021).

In practice, DT for cities is currently deployed to characterize different digital models and tools. (Deprêtre et al., 2022) propose an exploratory study of DTs for cities definitions and technical implementations. The authors present a comparative analysis between DT and CIM concepts since both terms are used interchangeably. The study's findings were based on the literature review of definitions and applications of the concepts as well as an exploratory survey of 13 practitioners about the potential of DT in the urban environment. To exemplify their findings, the authors shed light on some developed projects, either named CIM or DT.

These related works motivate us to list all the definitions of DT for cities and to conduct an in-depth study of definitions and related concepts, both from academia and practice.

2.2.2 Data integration challenges

To build a DT for cities, data integration methods are crucial (Lei et al., 2023; Shahat et al., 2021), especially since data are derived from disparate and heterogeneous sources and span different time and spatial scales. This is exemplified in the DTs implemented in practice, which integrate data from different sources such as data acquisition methods, existing geodatabases, real-time sensor data, actuators, crowdsourcing, etc. Many authors focus on the geometric aspects of data integration. They propose various methods for combining geometric information derived from various sources, whether related to 3D city models (Deng et al., 2021) as the input layer of the urban DTs (Bacher, 2022; Döllner, 2020; Lehner and Dorffner, 2020). The effective integration of heterogeneous data in DTs is not limited to the geometric part but also takes into consideration the semantics, the structure and the storage methods (Noardo, 2022). DTs need to integrate multi-source data within common system, and, for each use case (mobility, flooding, air pollution, etc.), the data from DTs must fit the specific requirements. In the field of geoinformatics, the topic of data integration was approached by extending the existing standard CityGML. It is considered as the most popular standard for integrating urban geodata for a variety of applications in the context of SCs and, more recently, urban DTs, with new properties. For common applications, the CityGML 2.0 model is augmented using the Application Domain Extensions (ADEs), namely Noise ADE (Kumar et al., 2017) and Energy ADE (Rossknecht and Airaksinen, 2020). Another example of the data integration model is a study established by (Chaturvedi, 2021) that introduced the concept of "Dynamizer" to model, store and exchange the dynamic variations of properties and time-series data implemented as ADEs for CityGML 2.0. Further work was carried out in the new CityGML 3.0 core module to consider the "Dynamizer" concept (Kutzner et al., 2020). The "Dynamizer" module is developed to enhance the usability and the integration of highly dynamic variations of properties whether provided from simulations or derived from sensors or IoT devices. The new "Dynamizer" module allows the creation of an explicit link between sensor/observation data and the respective city object properties within a 3DCM that they measure. In this sense, the "Dynamizer" concept provides a method to handle the dynamic properties of city objects with respect to the application requirements. In addition to the "Dynamizer" concept, the new version 3.0 of CityGML⁵ introduces various new features and improvements of existing modules that open up new applications (e.g., supporting point cloud, enhancing building and construction modeling,

⁵ <https://docs.ogc.org/guides/20-066.html>

managing various cities versions to name few) (Kutzner et al., 2020). Thus, facilitating data integration for SCs and DTs for cities. Furthermore, (Santhanavanich and Coors, 2021) proposed the concept of CityThings. Their approach explains how to handle sensor data stored in separate databases and associated to a 3DCM using SensorThings API and CityGML standards. The virtual model can be updated according to real-time updates based on Internet of Things (IoT) technologies. Nevertheless, the integration of sensor data in the DTs applications is part of the vast amounts of datasets (from GIS, BIM, organization databases, etc.) that can bring the virtual replica of the city closer to its physical counterpart. Furthermore, most of the data integration initiatives focus primarily on a specific domain application. For example, many cities around the world are showing an increasing interest in managing their building energy consumption and achieving energy reduction goals in the energy domain (T. Santhanavanich et al., 2022; Zhang et al., 2021). Up to date, various initiatives have addressed data interoperability and data integration challenges in the urban environment and SCs domains. For example, the FIWARE smart data model has shown its capabilities in harmonizing SCs data, namely sensor streams, for more than 200 cities worldwide (Bauer et al., 2021). The FIWARE open-source components use the NGSI-LD standard, which unifies and integrates the running platforms under the same standard data format (Cirillo et al., 2019). Accordingly, using FIWARE components provides a backbone for SCs data infrastructure. Furthermore, the study from (Bauer et al., 2021) shows that the NGSI-LD information model is used to create a DT infrastructure that represents the reactive and predictive DTs functionalities. The NGSI-LD information model allows the modeling of an Urban Digital Twin as entities. These entities have properties that should be modeled as well. Thus, using a property-graph-based model where entities are considered as nodes and relationships as edges. In addition, assessing and managing relevant data from DTs is also enabled using the NGSI-LD API.

Data interoperability is the core challenge of the OGC community, which focuses on ensuring clear semantics, fostering data modeling, and developing APIs to support application web development and facilitate data access and exchange. Thus, an overall reference architecture is defined, based on the review of OGC initiatives and standards⁶ to create an open and interoperable system of systems to make SCs and DTs a concrete reality (Atkinson et al., 2022). By analyzing DTs for city implementations, different approaches are used for data integration, data derived from different sensors and delivered from multiple services, mainly through databases that communicate with each other. For example, Vienna is linking databases to form a CIM that will be a basis for DTs as well as creating a central planning database; furthermore, relationships between linked databases and simulation results are expected (Lehner and Dorffner, 2020); at the level of the CityGML Schema (the extension of the conceptual model); or at the level of applications that disseminate several datasets in an integrated platform. DTs for cities allow the reuse of existing datasets through data integration. Up to date, a related work presented by (Lei et al., 2022) develops a holistic and comprehensive multi-level approach to assess and benchmark the open and accessible 3D city models, taking into consideration four categories (data portals, basic information, thematic content and attribute content), comprising 47 criteria. The study allows us to extend the framework with the data integration aspects that are relevant for implementing DT for cities. In practice, there has been no related work that classifies the DTs for city initiatives from the data integration perspective

⁶ <https://www.ogc.org/standards/>

2.3 Methodology

To meet our research goals, we start by gathering papers related to the DT for city definitions using various search terms (“**City Digital Twin**”, “**Urban Digital Twin**”, “**Geospatial Digital Twin**” and “**Spatial Digital Twin**”) conducted in Scopus. An initial corpus of 92 papers covering a wide range of fields is obtained. By refining the results and narrowing the scope to our field of expertise, namely computer science, urban planning and geospatial-related fields, 87 articles are collected. By performing more boolean operators to refine our results, eliminating the duplicate results from the list and examining the paper title, abstract and keywords, we gathered 29 articles. A specific inclusion criterion used in the scope of this work is that articles related to DTs need to be at the city-scale level (e.g., the entire city or district). As a result, 26 articles are maintained for in-depth analysis. To compare DT for cities with other concepts, we performed three specific queries for each analysis. The methodology deployed in Section 2.4 and the resulting number of papers are summarized, in **Figure 2.2**.

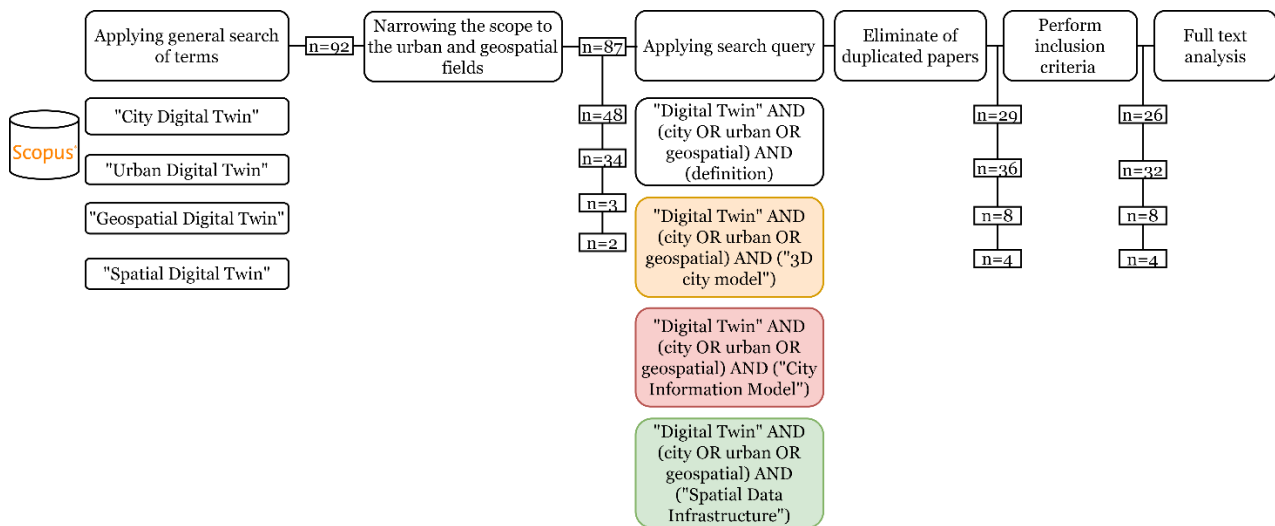


Figure 2.2. Steps and search queries to identify research articles about Digital Twin for cities in the urban and geospatial domain.

To document the current DTs for city implementations and inspect the data integration method, we collect scientific articles and reports related to each initiative to get an overview of the data, the methods and the technical frameworks.

Further analysis is carried out using the survey results to obtain an overview from scientists and practitioners about their definitions of DT for cities and its potential in their

field of expertise, as well as to specifically measure how faithful the practice is to the literature. The survey is organized as illustrated in **Figure 2.3**.

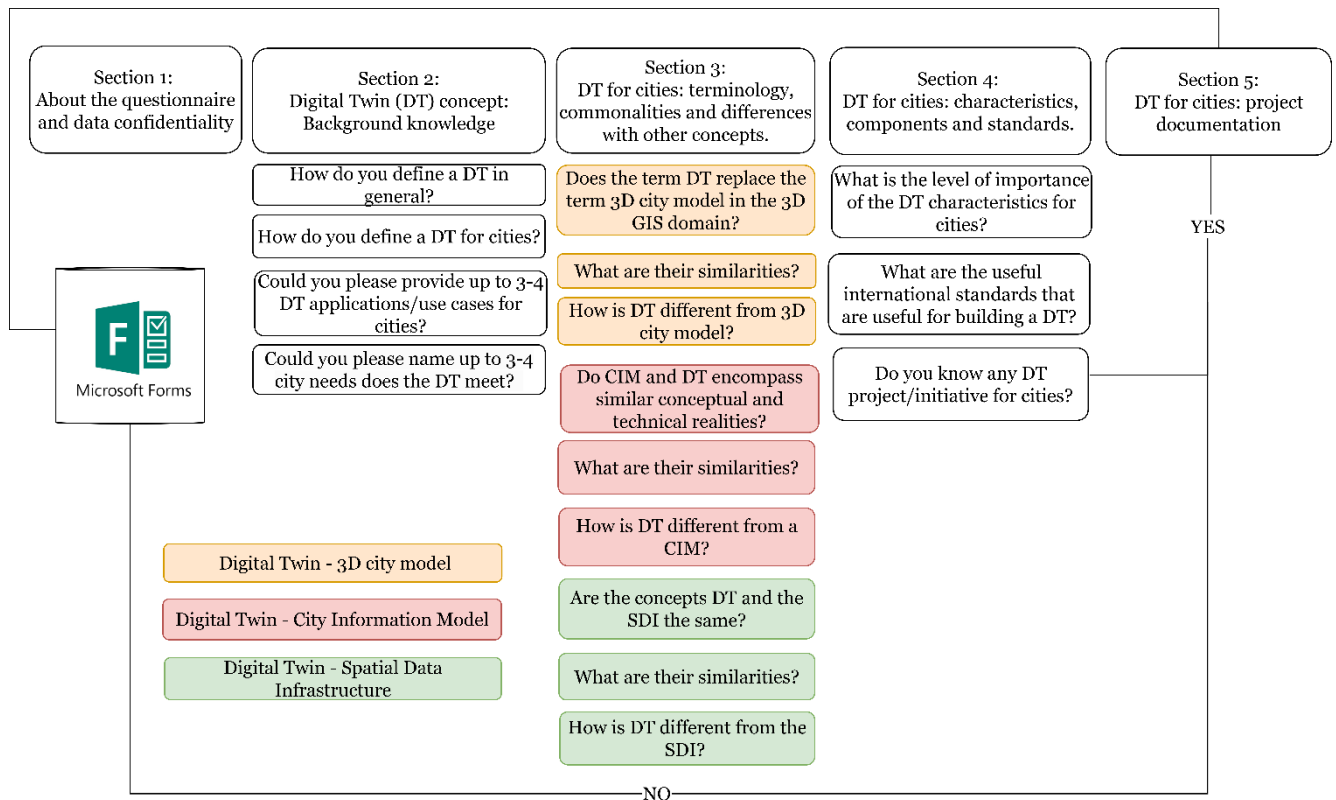


Figure 2.3. Survey sections conducted to complement the scientific literature.

Sections 2, 3 and 4 of the survey are used in the scope of the first results of this article. Section 5 is designed to collect information about DT projects for cities and to complement the various initiatives with the missing data regarding the technical implementations. We ask different participants from different sectors (e.g., academia, industry and government) to fill out the survey. The survey was sent to more than 30 members. As a result, 17 responses were collected and analyzed. The results are presented in this work in section 2.4.3.

2.4 Digital Twins for Cities: Analysis of Definitions and Study of Concepts

The term "DT" has been predominant in the literature in a transdisciplinary way. Many authors initiate their research by defining DTs for cities; however, these definitions present some differences that need to be studied.

So, the questions that arise are: **How to define a DTs for cities? What is different from what was done in earlier studies in the 3D modelling? Is it only limited to the interaction between real and digital models using real-time data?**

In this section, we first start by refining the theoretical-conceptual definitions of DT for cities presented in the literature, even if the research on implementing DTs for cities is still in its early beginning. We then explore similarities and discrepancies between some

of the concepts, since many previously discussed concepts are reused as the backbone of DT (Boje et al., 2020) and are sometimes renamed “DTs”. Some authors have reviewed the concept of DTs for cities with other approaches, likewise 3D city models (Ketzler et al., 2020) and city information models (Deprêtre et al., 2022). We then analyze the survey results following the same structure of section 2.4.1.

2.4.1 Digital Twins for cities: Systematic literature review of definitions and concepts

DTs for cities: Analysis of current definitions

It is unclear and ambiguous in the scientific literature how DTs for cities are conceptualized (P. Alva et al., 2022; Deprêtre et al., 2022; Shahat et al., 2021; Batty, 2018). This observation is made since some authors focus on city entities (buildings, roads, vegetations), some on urban infrastructure systems (transportation, water networks, electric power networks) and some on processes and services. This leads to many definitions coupled with specific characteristics that are adjusted for each city system. The lack of a common and universal definition means clearly that we are far from having a single shape of DTs for cities. To contextualize the concept of DTs for cities, we initially adopt a broad definition borrowed from the purposes of using the DTs in the urban settings and SCs. Indeed, DTs for cities are considered as a dynamic digital solution designed throughout its lifecycle to achieve sustainability and facing the increasing cities challenges (Hämäläinen, 2020): population growth, limited resources and climate changes, etc.

Reviewing the literature, we establish a list of the recent definitions mainly published between 2020 and 2022 for more mature understanding of the term (City/Urban) DTs. The definitions are presented in [GitHub](#) and are classified according to their document type, the year and their associated labels to name DTs.

A systematic analysis of the definitions allows us to identify three parallel tracks: **one track emphasizing that the core of DTs is a 3D city model** (Dembski et al., 2020; Schrotter and Hürzeler, 2020; Dimitrov and Petrova-Antonova, 2021; Bacher, 2022; Lehtola et al., 2022; Ferré-Bigorra et al., 2022; Lee et al., 2022; Bacher, 2022; Würstle et al., 2022; Ketzler et al., 2020; P. Alva et al., 2022). **Another track prefers to use DTs for cities as a natural convergence of BIM and GIS** (Cureton and Dunn, 2021; Xia et al., 2022). The third one prefers **the term DT to broadly describe a realistic digital representation of the city elements that function as a system of systems without narrowing the scope to the 3D component** (Lu et al., 2019; Papyshv and Yarime, 2021; Raes et al., 2021; Nohta et al., 2021; Agostinelli et al., 2021; Deng et al., 2021; Caprari, 2022; Nguyen and Kolbe, 2022; Lee et al., 2022; Hristov et al., 2022; Deprêtre et al., 2022). All tracks highlight that data, models, simulations and visualization technologies are the main components. The existence of these parallel tracks is justified by the involvement of different fields in urban settings.

Based on the definitions analysis, the three parallel tracks motivate us to put the DTs for cities into a more advanced analysis with the 3DCM, CIMs and SDIs to bring them meaningfully together. These concepts are used together in scientific literature and sometimes encompass similar conceptual and technical implementations.

A simple analysis of the definitions shows that most of the research has led to a tacit agreement of what constitutes a DT for cities in the geospatial domain and the SCs initiatives previously announced by (Stoter et al. 2021).

Indeed, the majority of the definitions clearly emphasizes the extending use of the (1) 3D city models enriched with semantic information, (2) often coupled with historical and sensor data in near or real time (depending on an appropriate rate of synchronization), thus enabling (3) a connection (e.g., data flow between the real counterpart and the virtual twin and vice versa), (4) allowing updates and analysis through a variety of simulations, predictions and visualization tools (web applications or game engines platforms) and offering (5) an integrated view of the multiple datasets, models through its life cycle allowing to understand and adapt city current and future states. This analysis enables us to evaluate some of DT's characteristics for cities along with other features which will be used to evaluate their degree of importance among practitioners through the survey.

DTs for cities: Key features and characteristics

As mentioned earlier, **data is the main pillar of the DTs for cities**. By unpacking the definitions used in this work, all authors mentioned the integration of heterogeneous data namely IoT and sensors data that allow a real-time bi-directional link between the two worlds (physical and virtual). Furthermore, some authors state that the historical data are also useful to implement a DTs. In addition, some authors have highlighted the relevance of geospatial technologies and reality capture in the creation of DTs for cities. Indeed, replicating the static city objects needs a geospatial data as input layer. Hence, some works are carried out to properly build a form of geospatial DT providing one of the promising foundation: the 3D static digital model of the city (Bacher, 2022; Lee et al., 2022; Lehner and Dorffner, 2020).

Some authors point out that a DT for cities requires a unique and generic digital platform. This characteristic of DTs is a little equivocal: few authors argue that a DT, as an exact real-time mirror of city, is unachievable since the DTs in SCs should consider the socio-economic components (Batty, 2018). Instead, many suggestions were discussed such as **“cyber-physical-social system with coupled properties”** (Charitonidou, 2022; Tomko and Winter, 2019), **“Social Urban Digital Twin”** (Yossef Ravid and Aharon - Gutman, 2022), or **“digital multiples”**, just to name a few. The “digital multiples”⁷ label proposed by Prof. Stoter comes from the point that an adapted “twin” is developed for specific applications and for different purposes. Each implemented twin in the best case would be systematically aggregated together, providing an ecosystem of twin systems.

Another particularity of DT, that is as well questionable, is the bidirectional connection between physical and virtual worlds. Based on the classification suggested by (Kritzinger et al., 2018) in the manufacturing field, three terms need to be defined: DT might be misused as a digital model or digital shadow. Based on the level of data integration from both digital and virtual worlds, (see Figure 2.4), a digital model (DM) is a simple abstraction of the physical object, any change in real world needs to be fed manually in the digital copy and vice versa. A digital shadow (DS) is when the data flows automatically from the physical object to the digital copy, but manually from the digital world to the physical entity. In contrast, DT is an automatic data update in both directions without human interventions. Accordingly, the data flows from the virtual copy to the physical world using artificial intelligence and actuators.

⁷ <https://www.intergeo.de/en/news/was-digitale-zwillinge-koennen-und-was-nicht>

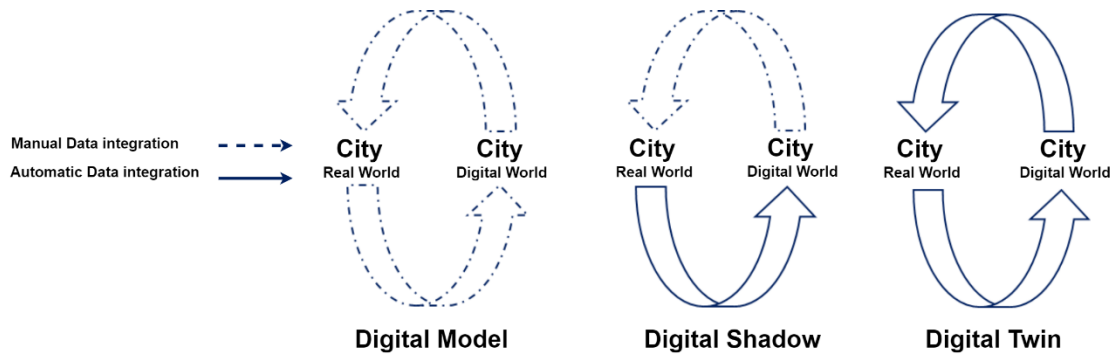


Figure 2.4. Digital Twin concept: Maturity levels according to the level of data integration.

DT for cities: Terminology

Regarding the terminology issues, the City Digital Twin (CDT) and the Urban Digital Twin (UDT) are jointly the most common terms used in literature. However, the terminological distinction is noteworthy. According to our analysis mentioned above, many authors use the term City Digital Twin to describe the 3D city model itself, e.g. a static replica of the city mainly focusing on buildings (Scalas et al., 2022; Hristov et al., 2022; Lehtola et al., 2022; Agostinelli et al., 2021; Nocht et al., 2021; Dimitrov and Petrova-Antonova, 2021; Ketzler et al., 2020; Schrotter and Hürzeler, 2020). Other authors use the CDT or DTC (Digital Twin Cities) to represent a dynamic representation of the city objects (e.g. a digital copy augmented with near real time data mainly traffic or air quality data) (Caprari, 2022; Deprêtre et al., 2022; Lehner and Dorffner, 2020; Raes et al., 2021). Others use the term Urban Digital Twin as a structure that can model multiple systems that could be potentially associated with the CDT as explained earlier (Dembski et al., 2020; Lee et al., 2022; Lehtola et al., 2022; Nguyen and Kolbe, 2022; Scalas et al., 2022). Hence, the UDT is a platform that uses the CDT as input and other modules (infrastructures, services and systems) can be added to extend the usability and the applications. The concept of system of systems introduced by the Centre for Digital Built Britain⁸, is the building block of the Urban Digital Twin that merge several twins in one common urban information model (Bolton et al., 2018). Geospatial Digital Twin or Spatial Digital Twin, they are used to emphasize the potential of existing geospatial data to meet the DT requirements. Thus, Spatial Digital Twin provides a spatial context of the objects, the infrastructures and the systems.

In the following part, a comparison with related concepts e.g., 3DCM, CIM and SDI is conducted to study the similarities and the differences with DTs for cities.

2.4.2 A comparative literature review with related concepts: 3DCM, CIM, SDI

Analyzing the three tracks of DTs definitions, DTs for cities have many overlaps and differences with other related concepts (3DCM, CIM, SDI) that need to be studied. This section presents the results.

DTs for cities and 3DCM

⁸ <https://www.cdbb.cam.ac.uk/>

3D city models gained significant interest in the past years and for a range of purposes. Historically, 3D city models helped to overcome the limitations of 2D objects representations (Herbert and Chen, 2015; Lehner and Dorffner, 2020). They were in many cases focused on the 3D detailed buildings modelling, considering buildings as the main features of cities (Corongiu et al., 2018; Döllner and Buchholz, 2005; Virtanen et al., 2021; Zhu et al., 2009). It gained high interest in 3D modelling field for several purposes (Chen, 2011; Kolbe and Donaubauer, 2021): energy consumption, city management, indoor navigation and many more.

Traditionally, 3D city models are usually created from existing 2D and 3D geodata sets generally acquired by aerial images (oblique photogrammetry) and LiDAR point clouds (Xue et al., 2020). To support the process of producing 3D city models, many geodata workflows are in practice that allow the public administrations and organizations to upgrade to 3D their geodata (Ledoux et al., 2021; Lehner and Dorffner, 2020). As a result, two models are in common use: photorealistic mesh models for visualization purposes. Such models require a full picture of the real scene with high geometry and texture (Adreani et al., 2022). Or in the case where the resulting model should serve as a basis for spatial analysis, a continuous semantic enrichment is recommended to extend the spatial and temporal scale. Thus, the application and the usability of the 3D city model are important (Billen et al., 2014).

3D city modeling is not only about data acquisition and processing, it is also inextricably linked to data management, storage and exchange. Hence, open standardized data model and exchange format for 3D city models are used. The CityGML and its lighter encoding CityJSON (Ledoux et al., 2019): are the most established data formats for 3D city models starting from simple representations to more rich and detailed information model according to the level of detail (LoD) required.

To better understand the interrelations between entities, infrastructures, processes and services in a city's dynamic ecosystem, silo-based models are no longer enough. Hence moving towards an integrated and maintained one stop-platform fed with heterogeneous data (Katal et al., 2022). This explains the shift from 3D static city modelling to more dynamic and alive representations. This paradigm shift along with the opportunities offered by the available digital tools gives the concept of DT something to demonstrate its potentials namely in the data storage and data management (Nys and Billen, 2021).

An initial strong focus is on 3D city models and its applications (Biljecki et al., 2015). Historically, (Ketzler et al., 2020) mentioned that the literature review used predominately 3D city models and only recently with the maturity of the digital technologies and the recognized value of the DT in other fields that the term DT for cities was gradually introduced from smart manufacturing to SCs.

The most popular manifestations of the DTs in the context of the geospatial field resides in the building or reuse of 3DCMs. With the help of visualization technologies, experts and non-experts could have a basis for communication and collaborative decision-making tools. However, a considerable number of articles focuses mainly on the geometric and semantic layer to implement the City Digital Twin (Diakite et al., 2022; Khawte et al., 2022; Scalas et al., 2022).

DTs for cities and CIM

Similarly to the previous analysis regarding 3DCM and DTs, the literature brings together CIM and DTs for Cities considering them as emerging technologies used to help designing future SCs (Omran et al., 2022). Both concepts are used interchangeably to address the

sustainability of cities. In the SCs scope, CIMs as well as DTs are conceived as a practice that takes advantage of digital technologies to assist urban planning and city management, to enhance collaboration between different stakeholders and to address the number of challenges that contemporary cities are experiencing with respect to the Sustainable Development Goals (SDGs) (Schaufler and Schwimmer, 2020; Tzachor et al., 2022).

CIM has appeared a decade ago to naturally extend the BIM and Computer-Aided Design (CAD) concept at the city scale (Xu et al., 2021). In the literature, various approaches are commonly used to describe CIM depending on their field of expertise, same as DT. Authors considered the CIM as an evolution of the traditional 3DCMs that go beyond visualization: a semantically enriched urban information model as a basis for a design and planning platform. Others focus on the BIM-GIS integration (Souza and Bueno, 2022) approaches widely discussed in the conversion framework from IFC to CityGML and its challenges regarding the full conversion (Zhu and Wu, 2022). Technically, BIM and GIS can be fused to enrich 3D city models. Furthermore, BIM offers the possibility to locally update 3D city models geometrically and semantically and, to increase their level of detail (Noardo, 2022).

Looking at the grey literature and starting from the mapping of the disciplines and the technologies that enhance the implementation of the conceptual framework of CIM proposed in the technology report⁹ produced by IEC SyC Smart Cities¹⁰ and ISO/IEC JTC1/SC41 Internet of Things and Digital Twin¹¹, it is apparent that the DT is a digital tool which is deployed to contribute to achieve the goals of the CIM along with various technologies and concepts (BIM, GIS, Planning Support System (PSS)).

In the literature, we can identify three main approaches, (see **Figure 2.5**) (1) **CIM and DT encompass similar conceptual and technical realities** (Deprêtre et al., 2022); (2) **CIM is the first input layer for the development of a DT providing an integrated foundation for semantically enriched models build by gathering heterogeneous data from various sources: organizations and public administration databases, companies, citizens etc.** (Cureton and Dunn, 2021; Ferré-Bigorra et al., 2022; Ketzler et al., 2020; Lehner and Dorffner, 2020; Petrova-Antonova and Ilieva, 2021); and (3) **DT is a specialization of CIM and a support technology for implementing the conceptual framework of CIM** (Shahat et al., 2021).

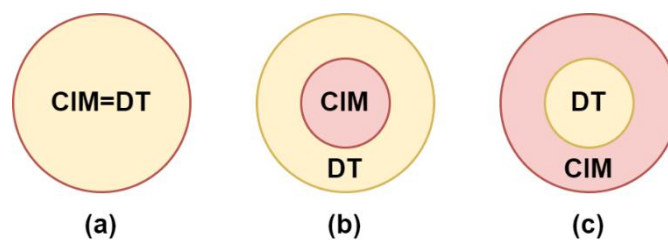


Figure 2.5. DT and CIM approaches: (a) CIM and DT overlap the same conceptual and technical realities; (b) CIM is the input layer for implementing DT for cities; (c) DT is a specialization of CIM and a support technology for developing the conceptual framework.

⁹ <https://www.iec.ch/basecamp/city-information-modelling-and-urban-digital-twins>

¹⁰ https://www.iec.ch/dyn/www/f?p=103:186:::::FSP_ORG_ID:13073

¹¹ <https://www.iso.org/committee/6483279.html>

As in the literature, there are different approaches that broadly identify the differences between the CIM and DT (Deprêtre et al., 2022; Omrany et al., 2022; Schaufler and Schwimmer, 2020). Since they are both theoretically and technically under the same umbrella (similar applications, technologies and objectives), a review of the SCs projects indicates that most of the implemented solutions of DTs are initiatives based on a minimum amount of CIMs that are more multi thematic, multi-scale, multi-temporal datasets and multi-actor approaches of cities.

DTs for cities and SDI

Considering city as a system of systems, huge amounts of data is generated, especially geospatial data, from various sources. Data are the core of creating and maintaining DTs for cities. Therefore, everything comes to data integration. The integration is not limited to the scope of linking heterogeneous data, but also to process, to store and to exchange them. Naturally, the Spatial Data Infrastructure (SDI) was built as a common basis to increase the availability and to foster interoperable access to geospatial data provided by multiple agencies and to allow the exchange of the data through harmonized and standardized spatial interfaces (Hu, 2017). The European INSPIRE¹² directive is one of the promising initiatives to implement an international spatial data infrastructure that addresses the technical and non-technical challenges of geospatial information across various levels. Thus, spatial data infrastructures for DTs may form a framework in establishing interoperability between systems and platforms in a standardized way (Kanishk Chaturvedi et al., 2019; Chatzinikolaou et al., 2020; T. Santhanavanich et al., 2022). The approaches based on SDIs have been discussed in the development of the DT of Zurich (Schrotter and Hürzeler, 2020) and in a recent work established in the energy domain (T. Santhanavanich et al., 2022). To manage the building energy data, the authors propose an SDI of a UDT platform using OGC standards, namely, OGC Sensor Things API, OGC API 3D GeoVolumes, OGC CityGML, OGC API features and Web Map services. The conceptual SDI for energy domain UDT consisted of four major layers: data source, data processing, web services and client application. The proposed SDI framework is an illustrative instance that might be applicable to any domain.

As a backbone framework to build a DT for cities, SDI will provide seamless access to data repositories (multi-scale, multi formats and multi-sources). Furthermore, a well-established SDI based on standards will facilitate the integration, the maintenance and the update of the various datasets to serve many applications.

2.4.3 Digital Twins for cities: Survey analysis of concepts and key features

To enhance our finding from the literature review, we designed an exploratory survey to: (1) refine the definition of DTs for cities and their maturity levels, (2) fill the gap between the theoretical concept and what has been developed so far, (3) investigate the evolution of the DTs initiatives that are implemented by scientists and practitioners and deployed for a wide range of applications and (4) discuss one of the omnipresent challenges that the topic is facing.

We have designed the survey¹³ (of five sections) in a logical succession of questions starting from a general definition of DTs and then narrowing the scope to the urban

¹² <https://inspire.ec.europa.eu/>

¹³ <https://forms.office.com/e/Yir5EacgUW>

settings and implementations to give the participants the possibility to answer, independently and according to their expertise and eventual involvement in some practical initiatives, the questions related to the theoretical part (section 1 to 4) and the one related to the practical part (section 5). The questions are typically derived from the main issues reported in the scientific literature and the challenges that the DTs for cities are facing. We designed the survey to confirm the results of the scientific review with practice.

To reach the survey objectives, we established a list of potential participants and a short introduction explaining the aim of the survey. Moreover, the selection of participants was not made randomly, it was based on their interest in implementing DTs for cities. They are identified as proficient in the field of DTs for cities and close-up domains. Indeed, we sent the questionnaire to more than 30 people via emails and LinkedIn.

Regarding the profile of participants, we asked participants about their background, their field of expertise and their degree of familiarity with the concept of DTs (Refer to section below).

Given the goals of our questionnaire, we expected from the participants that fill the form to get alignment about how they define DTs and its applications in the urban context and how they implemented in practice by (1) defining the DTs according to them, (2) listing their applications of DTs and their city needs, (3) understanding their vision regarding DTs comparing to other concepts that they use jointly in their definitions and their implementations, (4) documenting their own initiatives following the questions designed in the survey and, (5) fostering participants reflections while answering the survey to review the way in which they have defined and developed the concept.

Profile of participants

The survey was conducted with 17 international practitioners about their views and experiences on DTs for cities regarding the definitions and the characteristics, the applications, the commonalities and differences with close approaches. We asked participants about their backgrounds, fields of expertise and degree of familiarity with the concept of DTs.

A total of 14 international practitioners completed all the survey sections and for whom the resulting responses are analyzed. The profile of the participants is summarized in **Figure 2.6**. Participants are mainly from academia (universities or research institutes) and from different fields of expertise, namely, urban modelling, geoinformatics, building and city analysis, GIS, etc.

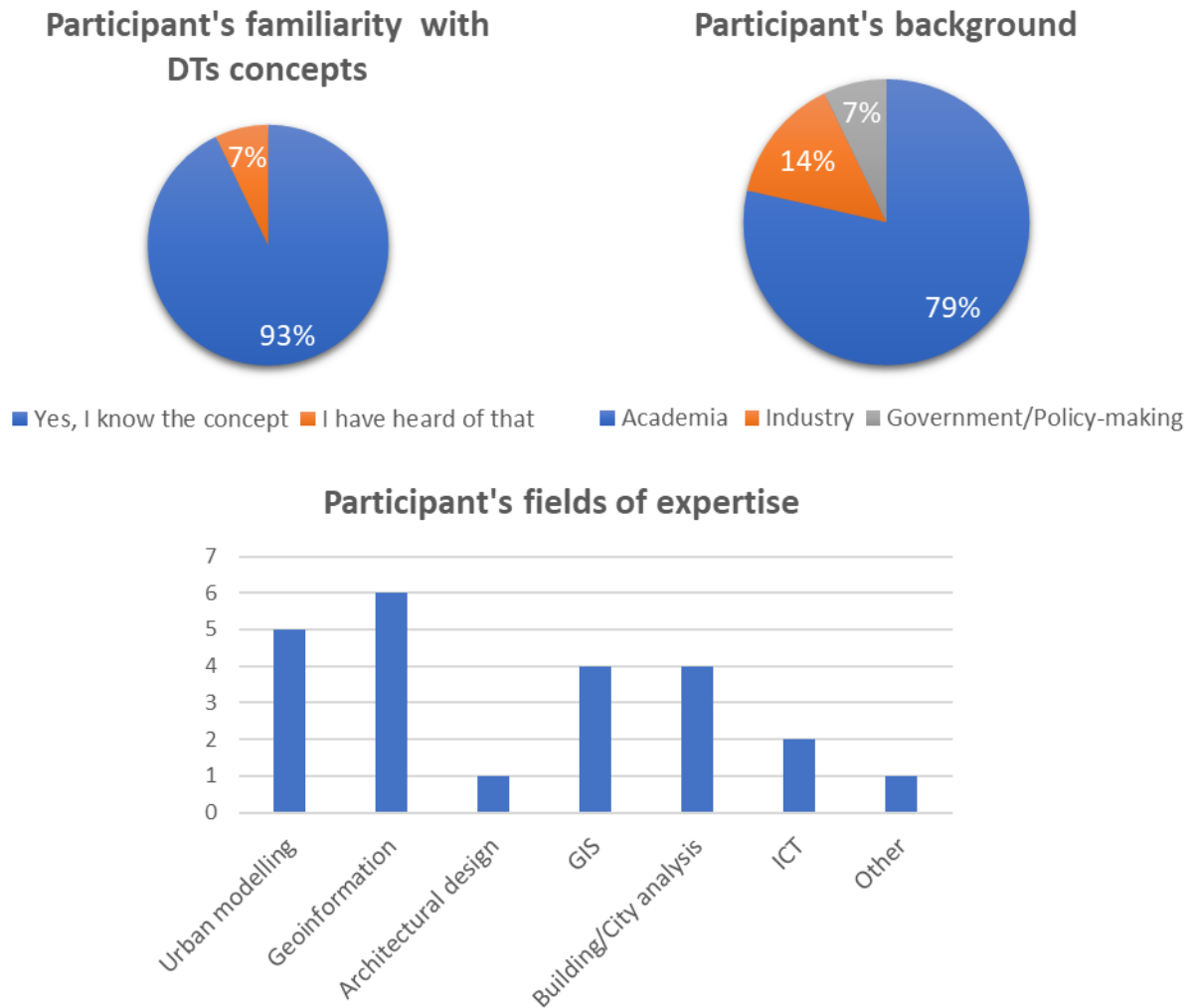


Figure 2.6. Participant's profile

Digital Twins for cities: Definitions, Characteristics and Applications

Given the analysis of the definitions derived from the literature and to understand how practitioners conceive the DTs for cities, the first section of the survey was about the DT definitions, applications and purposes. Most of the participants consider DT for cities as a living (digital/virtual) (representations/models) of the city existing real-world objects, infrastructures, systems, processes. A DT has a two-way data flow from the “virtual” (V) and the “physical” (P): data going from P to V and information going from V to P to fit the purpose in some specific use cases. The data flow allows the convergence between the two counterparts at an appropriate rate of synchronization (generally near real-time) through an ongoing process of enrichment with dynamic data collected from sensors and IoT devices. DTs for cities take advantage of reality capture, mapping and surveying technologies to build a virtual copy of the city. It is considered as a system of data, models and algorithms that continuously refines the digital representations.

To highlight the characteristics of DTs for cities, we asked the participants about the level of importance of the following DT features for cities. (See Figure 2.7).

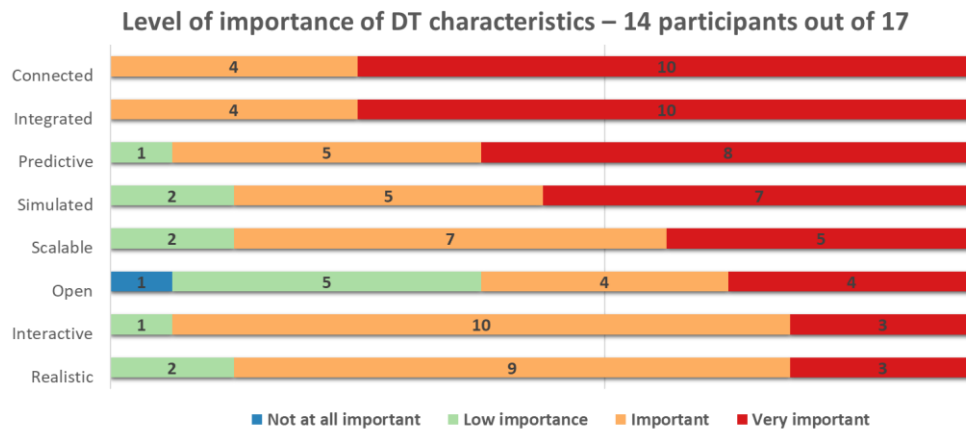


Figure 2.7. The level of importance of DT characteristics according to the survey.

The results show strong interest in the connection between both physical and digital worlds and consider that the integration aspect is one of the promising characteristics of a DT. Furthermore, the necessity of predictions and simulations is one of the main motivations towards DT for cities. DT for cities needs to be scalable as well. The realistic representations of the real world are crucial for visualizations purposes. However, relevant real-world abstractions are more important than having a realistic copy, so analysis and simulations could be conducted on the top of the virtual model.

During the survey, participants were also asked to provide some DT applications and use cases for cities, since DT for cities implementations are always related to the application. Most of them name urban planning, environmental management, traffic monitoring and urban mobility, disaster risk management, noise and air analysis and simulation, energy demand analysis. Three practitioners from academia: (1) ICT, (2) urban modelling and geoinformation and (3) architectural design backgrounds identify respectively some specific applications to support their city needs, namely: 15-minute walkable cities, real estate, heritage conservation and documentation.

DTs for cities and 3DCM

Recently, it has been largely noted that 3DCMs and DTs for cities go hand in hand in scientific literature and in different worldwide SCs projects.

The main question is: **how much this term DT is starting to replace 3DCMs as the predominant term in the 3D city modeling and SCs following the work presented by (Ketzler et al., 2020)?**

10 participants from different backgrounds found that the DT did not replace yet the term 3DCMs (see Figure 2.8). They argue that the 3D city models are not yet a mature version of DTs that take on board all the previously mentioned features. In addition, they brought to light the spatial and temporal scale of the DTs (e.g., DTs are more towards a general spatial and temporal scale than the 3D city models). They also emphasize the smartness, the dynamicity and multipurpose usage features of the DTs. Two participants from both academia and industry elaborate their answers by explaining that the DTs can be based on 2D data and that are several applications of 3DCMs which do not fit the theoretical definition of the DT but still could be useful for traditional visualizations purposes.

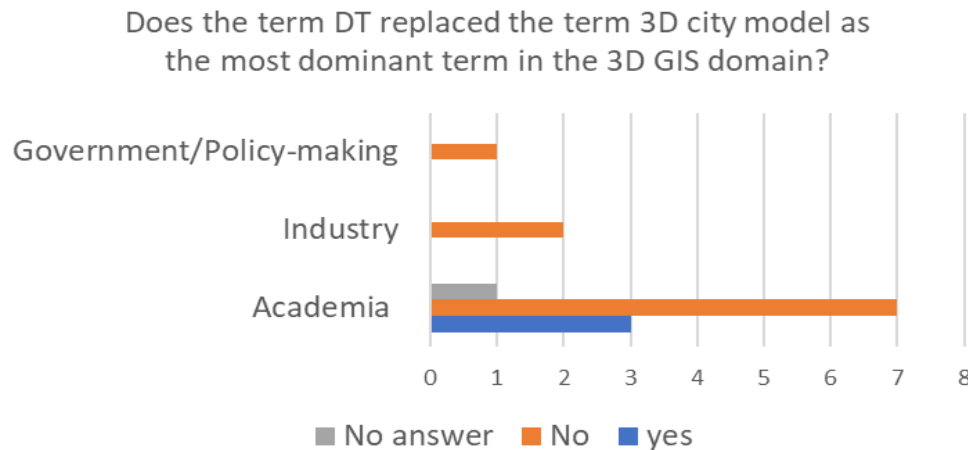


Figure 2.8. Differentiation of the levels of consent according to the participants' profiles about the concepts of 3DCM and DTs.

Reviewing the literature and taking into consideration the answers collected from the survey, considering the 3D city models as the basis to build DTs for cities is commonly accepted by practitioners to implement their DTs for cities initiatives from a geospatial point of view. First because of the availability of 3D models. Second, thanks to the historical achievements of 3D semantic models in performing a wide range of simulations and analysis. Finally, the other motivation is the fact that DTs enhance the collaboration between different stakeholders and tend to engage the citizens as end-users in the loop. This demonstrates the value of the 3DCMs for visualization purposes and communication with non-expert users.

To sum up, DTs and 3DCMs have a lot of similarities if we retain the idea that 3D city models are the building blocks of DTs. The latter should be connected, integrated, predictive, simulated and scalable. Moreover, DTs could learn from the technical challenges that the 3D city models have experienced in the decades of research in 3D GIS.

DTs for cities and CIM

To meet the literature statements, we wanted to have an overview from different participants about the similarities and the divergences between these concepts. Participants argued that a DT uses data that are collected and organized in the framework of the CIM. They also almost agreed on the statement that the use of CIM appears to be a starting point to build a City Digital Twin. 6 participants (4 from academia, one from industry and one from government) totally agreed that DTs for cities are a specialization of CIM.

7 participants (4 from academia, 2 from industry and one from city government) disapprove of the idea of considering CDT as a full integration of GIS and BIM, they consider that it could be a part of building and maintaining the city DT, but it is not the main core of conceiving DT for cities (see Figure 2.9). Almost, all participants explain that the IoT and real-time coupling properties and the two-way link (data-information) from both worlds are better handled and more comprehensively described in the literature when working with DT than CIM. CIM covers the standards that could be useful for the CDTs and may simply be a static representation of city objects that does not change continuously. In addition, DTs encapsulate more algorithms for analysis and simulations and use different standards and structures.

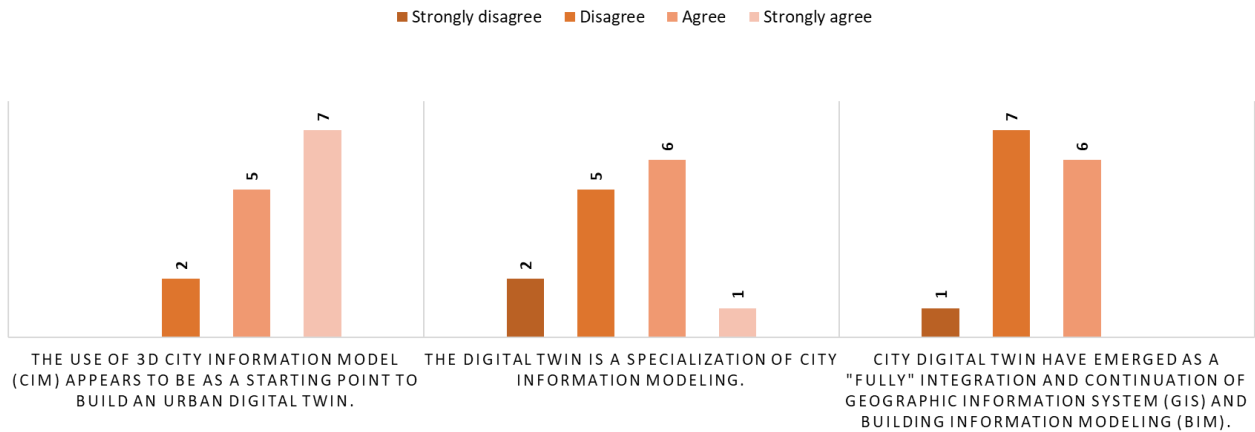


Figure 2.9. The level of agreement regarding the statements related to CIM and DTs.

To conclude, we can state that DT and CIM have both something to bring to the urban environment. Both need to have well established architecture, clear semantics, so future research can be based on, to deal with complexity of cities and understand the overlapping between the technologies.

DTs for cities and SDI

By looking at the survey response, almost all the participants agreed that solid spatial data infrastructure is the foundation for a UDT built on standards and interoperability, enabling effective usage and sharing of spatial information. However, 8 participants (7 from academia and one from industry) disagreed with the statement that DT is a spatial, digital model that extends the existing spatial data infrastructure with 3D spatial data, additional attributes and properties (see Figure 2.10).

In short, UDT is a spatially enabled platform that incorporates intrinsically the concepts involved in the SDI (policies, organizations, technologies, data and people). SDI gives an overview of how data are acquired and represented. The simulations, the predictions, the analysis and the linkage between the real and digital worlds are not covered in the SDI. The dynamic aspect is lacking as well, in contrast with DT. However, SDI is a foundation enabling the effective usage and sharing of spatial information.

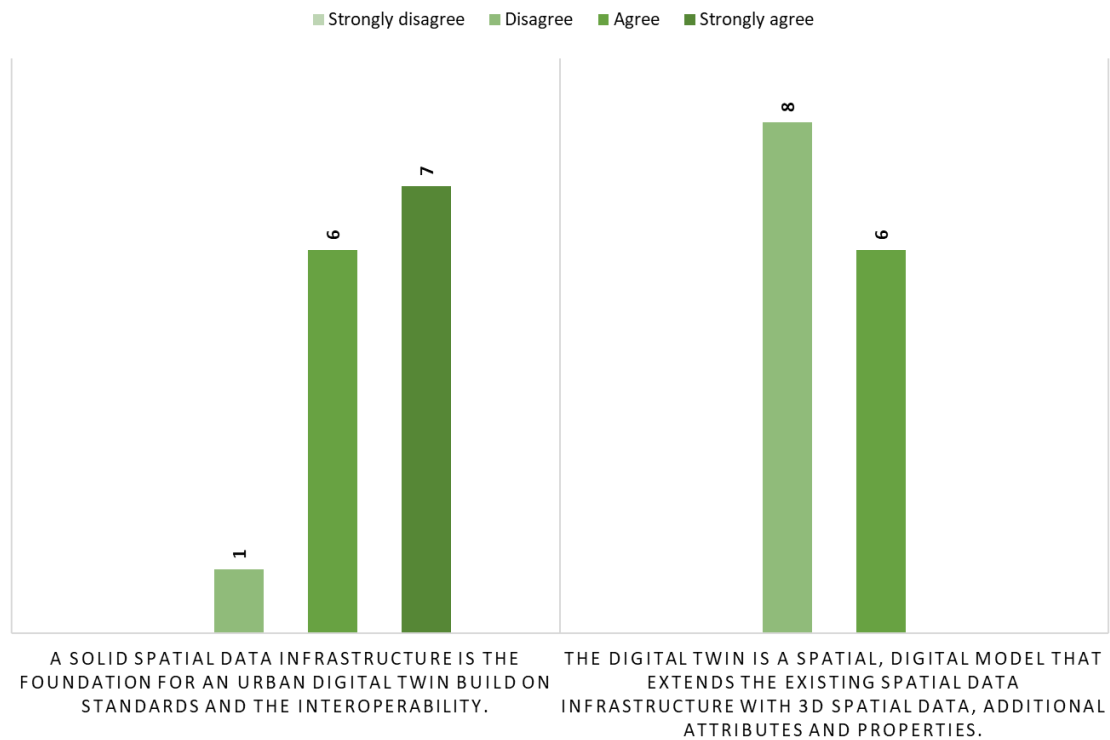


Figure 2.10. The level of agreement regarding the statements related to SDI and DTs.

2.4.4 Summary and discussion of the DTs definitions and concepts

According to the state of the art, there is no common definition for what DT for cities looks like. The insights generated from this first section provide a more in-depth understanding of the concept of DTs for cities. Thus, the DTs concept for cities is expanded to take the 3D digital city models from static models to more dynamic representations, from individual or blocks of city entities (e.g., buildings or roads, vegetation) to large-scale coverage (the entire city and its interdependencies synchronized at specific frequencies with respect to an appropriate level of fidelity), from a silo-based approach to a participatory-based approach. This definition summarizes the three major tracks identified in the literature. The different tracks motivated us to conduct an in-depth analysis of DTs and related concepts. Study of concepts enables us to clearly discern terms and concepts that are used jointly in literature as well as in practice. The results will help us understand how practitioners use DTs and under what umbrella (3DCM, CIM, SDI). Concerning the potential of city-scale DTs, many authors discuss deeply the key drivers and the advantages of the DT for cities (Hämäläinen, 2021; Ivanov et al., 2020; Mohammadi and Taylor, 2017; Shahat et al., 2021), although this is done mostly to focus on demonstrating the technical applicability of the concept without having a clear definition in mind. This digital replica of the city has many features and characteristics that set it apart from other digital models, specifically the connectivity and integration aspects. The connection feature of DT is defined by the mutual interaction between the digital copy and the real world. On the basis of this bi-directional data connection, we can identify 3 concepts: DM, DS, and DT.

To tackle the second major contribution of this chapter, we will take advantage of the definitions and characteristics taken from the first section and classify the initiatives according to their level of maturity and the level of data integration. We aim to not only list the relevant initiatives according to some explanatory categories but also provide a

critical and comparative approach to evaluate the DT initiatives. The main goal is to outline the different levels of data integration. These differences are justified by the diversity of the available data, the methods, and the use cases.

In the following sections, the results are presented from a technical perspective and according to some specific inclusion criteria.

2.5 Digital Twin for cities: description, analysis and classification of current implementations

The concept of DTs for cities has continued to be appealing in numerous cities around the world and attract considerable attention among both academia and practice (i.e., industry, governments and public administrations). The term is applied to a specific district or the entire city-state. Indeed, most of the initiatives are largely driven by purpose (i.e., to support the administration to solve a specific phenomenon and to help in climate and energy transition more specifically). The development of the DT started with proof of concept according to the available data. Hence, 3D city modeling gained significant interest in the framework of DTs for cities as one of the building blocks. Therefore, we are far from having a single form or output of DTs for cities. The heterogeneity of the technical frameworks is the result, firstly, of the lack of a common definition, as discussed in the previous sections. Secondly, the motivations towards creating DTs are different from one city to another. Finally, not all cities have the same data open or available to create DT, hence the availability and the quality of data are one of the prominent challenges that DT for cities is facing.

Reviewing the state-of-the art of DTs for cities, the most relevant city-scale DTs are built in a similar way (3D city models). It leads to this question: **how close are the theoretical definitions of the DTs to the technical perspectives?**

The main objective of this section is to study the current DT implementations and to get an overview of the ongoing work for DTs for cities since different cities have particular needs and various requirements. For this purpose, we include recent papers and links to various projects, prototypes and proof of concepts. We also used the survey results derived from section 5 of the questionnaire.

To facilitate the analysis conducted in this chapter, we first propose categories to document the DT for city initiatives (refer to Table 2.1), then we summarize general information about the DT initiatives (see Table A.1). We then classify the DT initiatives according to their level of data integration (see Table 2.2). In this sense, we propose three levels of data integration from a geospatial perspective (see Table 2.3). We conclude by summarizing the findings.

2.5.1 Digital Twins for cities initiatives: Study of realizations

The state-of-the-art is not well-developed regarding how the DTs are implemented in practice and does not provide an in-depth analysis of the technical requirements. Only a few scientific papers cover and present the recent DT initiatives. Hence, in the scope of this section, the selected digital cities replicas are presented, under the condition that they are well documented on their online websites or reported in one or more scientific articles.

In this section, we define some inclusion criteria to filter the DTs for city initiatives that will be analyzed in this work. First, the scale should be at the city or district level. Second, the DT has a 3D city model as input data. Third, the DTs are conceptualized to apply

further analysis and simulations on top of visualizations. While it is outside the scope of this article to list all existing DTs since some initiatives could be confidential and the data collection from the grey literature could be overwhelming. An overview of some of the most ongoing projects around the world is discussed in more detail and for which the necessary information is available.

To better present different projects, we proposed the following categories (see **Table 2.1**).

Table 2.1. Description of categories to document the DT for city initiatives.

Categories	Description
Scale	Coverage of the city project. The initiatives are implemented at several spatial scales. We take into consideration the city, the district and the precinct scales.
Data sources	Summary of data acquisition, sensors and methods.
City models	Indicating the input layer of implementing DT (i.e., 3D city model, mesh model, hybrid model, or point cloud model). In this category, we specify the modeled city objects and the standards.
LoD	The level of detail of the city objects
Purposes	Explaining the purposes of creating DT for cities. The purposes are strategic and operational to meet the city's needs.
Visualization platforms	Several platforms are deployed to disseminate the DT contents namely, web user interfaces and game engines.
Simulations experiences	To list the simulations and the analysis performed using the DT for cities.
Update method and frequency	Maintaining and updating the DT for cities is one of the promising steps in the lifecycle of DT. Hence, some initiatives have their own methods of updates and define a framework for maintaining their data and models. The update frequency is also relevant in our research to have an idea about the age of the 3DCMs.
Status	The maturity level of development of the DT project: proof of concept/prototype- operational- under improvement (new features).

In APPENDIX A, **Table A.1** presents digital twin initiatives developed through multidisciplinary collaboration between universities, research institutes, and public and private organizations, categorized according to the previously defined criteria.

Analyzing the findings, DTs for cities are spatially enabled platforms. They are based on various datasets and predominately tend to use the hybrid method from different sensors data (aerial acquisition, laser scanning). Most of the initiatives are not built from scratch, they are based on a minimum core of CIMs or implemented based on existing SDIs.

Table A.1 serves as foundational information for how the data are acquired, how they are modelled and how data are visualized. An overview analysis shows that DTs for cities are developed in various spatial scales. It gives the possibility to go from small proof of concepts to large scale implementations. The most common use cases are related to energy domain, mobility and environment issues. We can also see that the use cases are generally created using the same data input, which explains the interest of having a full image of the city with all its objects, processes and systems. The more the model is rich semantically and valid geometrically, the more use cases can be conducted. For the level

of details, some DT projects combine different levels of details to serve multiple uses cases. However, for buildings generally, the DT initiatives use LoD 1.x and LoD 2.x as a simple model to manipulate with respect to the data available. The viewer generally provides the textured and untextured buildings. More detailed buildings can be integrated if BIM (as built) models are provided by CAD industry.

As mentioned earlier, buildings are the most omnipresent themes that are modelled, since they represent the identity of cities. Many initiatives express the need to incorporate other themes in the context of the DT to cover all the city objects and replicate all the systems. The visualization platform is commonly built on web-based applications using CesiumJS¹⁴, for dissemination of several datasets. Moreover, Virtual City Systems¹⁵ provide innovative solutions that render 2D and 3D geodata using web-based applications based on Cesium (i.e., VC Maps¹⁶). Other initiatives explore game engines capability, mainly Unity¹⁷ and Unreal Engine¹⁸, as a platform for visualization and interactions. From an academic point of view, Cesium has shown growing popularity in the DT context through its intuitive development and visualization functionalities. However, from practice, ESRI software packages¹⁹ namely ArcGIS Online, ArcGIS Urban, JavaScript for ArcGIS (using I3S) and ArcGIS Maps SDK for Unity and Unreal Engine are widely deployed as a foundation for DTs. Given the complexity of creating DTs for cities, some considerations are identified in the ArcGIS requirements namely, time scale, stakeholder diversity, systems complexity, data ownership and data security.

Generally, simulations can be conducted as well using the game engines (i.e., traffic and rainfall simulations in Virtual Gothenburg). The DTs for cities are based on open data standards (i.e., basis GIS data standards, OGC standards, ISO standard for metadata). The most common 3D city data model is CityGML model. For the moment, there is no active initiative that relies on the CityJSON format, which defines itself as more “developers friendly”. All attempts to build DT are based on CityGML 2.0, which explains the differences noticed in the data integration frameworks (integration of dynamic data and versioning). These issues should be answered with the latest version 3.0 of CityGML as long as better integration (BIM-GIS) that is one of the update methods for DT for cities context. Some initiatives offer the possibility to render different formats of 3D models in the same viewer (mesh model, 3D point cloud, hybrid model, etc.). It is also worth mentioning that most of the simulations are performed with separate software and the results are integrated using different approaches (refer to section 2.5.3). Some of the projects specify their methods and the frequency of updates. Many initiatives are operational. They tend to provide a collaborative digital platform to understand the efficiency of current systems and to design and test future scenarios. The list of DTs initiatives is not exhaustive, and further projects could be documented for future articles.

¹⁴ <https://cesium.com/platform/cesiumjs/>

¹⁵ <https://vc.systems/en/>

¹⁶ <https://vc.systems/en/products/vc-map/>

¹⁷ <https://unity.com/solutions/digital-twins>

¹⁸ <https://www.unrealengine.com/en-US/digital-twins>

¹⁹ <https://www.esri.com/arcgis-blog/products/arcgis/aec/gis-foundation-for-digital-twins/>

2.5.2 Level of data integration between both worlds.

Taking insights from APPENDIX A and considering the definitions of the DM, DS and DT, we classify DT initiatives of cities according to the type of digital replica (static or dynamic i.e., include the sensors or IoT data) and the level of data integration (data connection between two worlds). **Table 2.2** summarizes the findings.

Table 2.2. DT initiatives classification according to the data type and the maturity level of integration.

CDT	Type of DT	Maturity level of integration
Helsinki 3D+ Kalasatama	Static	Digital model
Espoo	Static	Digital model
Vienna	Static	Digital model
Rotterdam 3D	Dynamic (near real time data, real time data)	Digital shadow
Zurich 3-4D	Static	Digital model
Virtual Singapore	Dynamic (real time data)	Digital Twin
Amsterdam3D	Static	Digital model
Digital Twin Munich	Dynamic	Digital shadow
Rennes 3D	Dynamic	Digital shadow
Virtual Gothenburg	Dynamic	Digital shadow
Herrenberg	Dynamic (near real time data)	Digital shadow
Sofia- Bulgaria	Dynamic (Near real time data)	Digital shadow
DUET	Dynamic	Digital Twin
Fishermans Bend Digital Twin	Dynamic (Near real time data)	Digital Twin
NSW New South Wales	Dynamic (Near real time data)	Digital shadow
DTV Digital Twin Victoria	Dynamic (Near real time data)	Digital shadow
Liveable City Digital Twin	Dynamic (real time sensor feeds visualizations)	Digital shadow
NUS-FRS	Dynamic (real time data)	Digital shadow
EnSysLE project	Dynamic	Digital shadow

All initiatives use the term “DT” in practice. However, according to the analysis of concepts and implementations, we are far from a complete implementation of the concept as it is defined in the theory. **Table 2.2** presents the different initiatives that are generally dynamic copies of the real world using the near real-time data collected from sensors. Based on the findings achieved in the section 2.4, we can conclude that almost all initiatives are digital shadows because the data connection from the real world to the digital copy is automatic. Still, the other way is generally done in a manual process (human interventions adapting the physical world). Such a bidirectional connection needs to be

explored in the future. The mutual connection developed up to now has not been achieved in any implemented city-scale DT. Since it will be costly and time-consuming to install an unlimited number of sensors and IoT devices in large-scale coverage. However, some works are carried out on a neighborhood scale to test the design of the DT through actuators and using artificial intelligence by giving feedback to the physical world. Thus, it would be preferable to name the current realizations Digital Shadow instead of the Digital Twin which is considered as a long-term perspective and that is not an end in itself, since new digital technologies will emerge in the future.

2.5.3 Data integration methods-based use cases

In this section, we intend to explore how data is integrated into DTs and assess the different data integration levels. We have classified the initiatives according to their use cases. The data can either be static data, dynamic data, or simulation results. To facilitate the understanding of the different levels of data integration involved in this work, we have defined the following schema (see **Figure 2.11**).

From a general point of view, we define three levels of data integration: a conceptual data model level (the data are integrated into the top level of the DT, i.e., extending the schema model to cover new features or themes), which will allow a direct feed of the database; a database level (data are integrated in order to feed or update specific attributes or classes; a conversion process is to be considered, particularly when data are in various formats); and an application level (the data are integrated into the application generally at the viewer level). The levels proposed in this work are inspired by how data has been integrated into 3D city models, especially those based on CityGML 2.0.

However, we have attempted to cover initiatives that use different standards without being limited to any particular standard. Indeed, there are many standards deployed to implement DT for cities, namely CityGML and buildingSMART IFC. These standards are recognized as the cornerstone of urban DTs. However, they address different applications and serve heterogeneous purposes as they have different and complex geometries, semantics, and structures and target different spatial scales. CityGML represents the semantic 3DCM according to a hierarchical decomposition of the city elements into relevant classes with respect to the level of detail. However, BIM as a natural evolution of CAD industry aimed to create 3D digital models of buildings or infrastructure by supporting their entire life cycle, ranging in different levels of representation, and enhancing collaboration between various disciplines. BIM models are generally more detailed representations incorporating semantics that are relevant for some specific applications (i.e., refurbishment). In parallel, 3D semantic models have higher-level semantics that allow performing various applications on a city-scale.

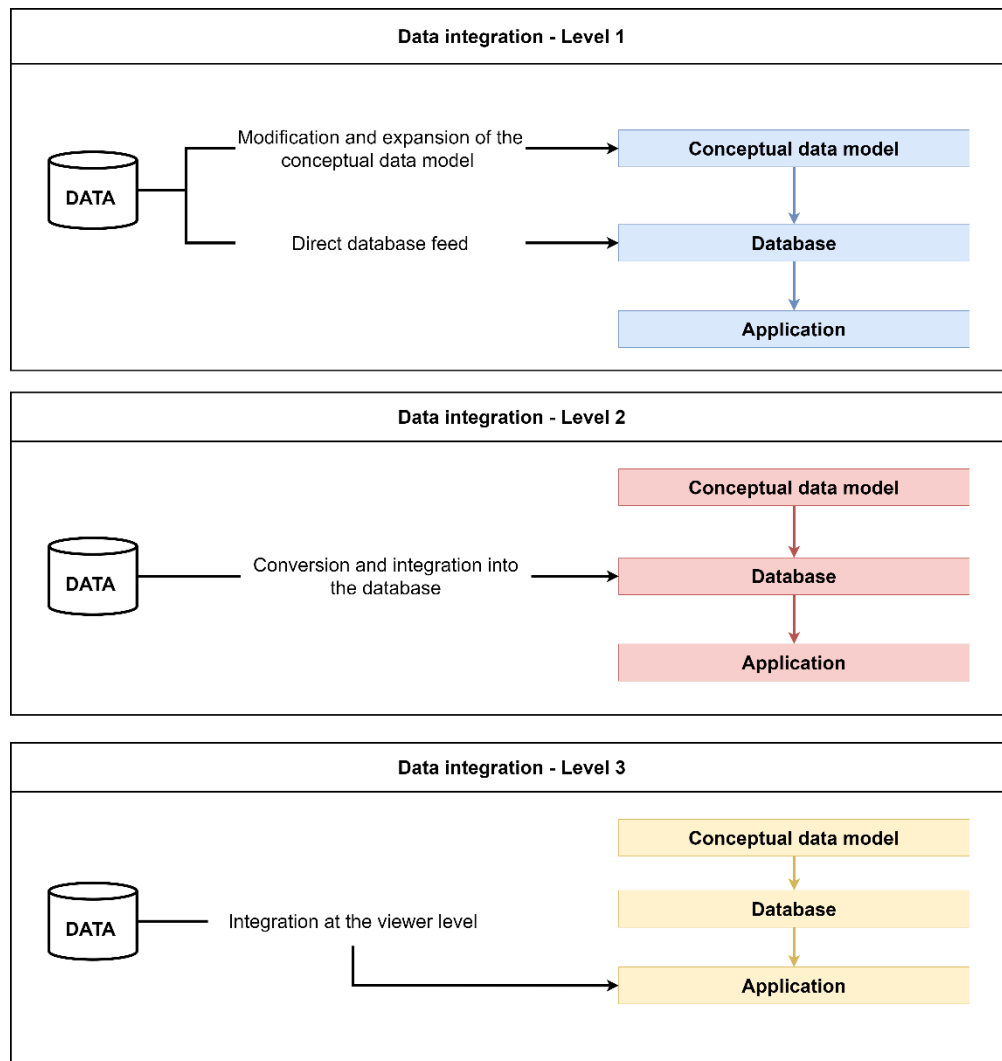


Figure 2.11. Levels of data integration: Level 1- Conceptual Data Model, Level 2- Database, Level 3- Application.

It is worth mentioning as well that 3D city models are initiatives that are dominated by research institutes and government, while BIM models are built and maintained mainly by industry. Thus, fostering the involvement of different stakeholders towards the use of both models in the city's scope. 3D city models and BIM models both have a great input to enhance the 3D city modeling and analysis within different applications from small scale to large-scale urban environments. However, the integration of both models is still challenging when it comes to the full conversion between both systems due to their geometry complexities and semantic heterogeneities. Hence, many methods and projects are carried out to perform uniform and standardized conversion of BIM data to GIS data and vice versa (GeoBIM initiative). Accordingly, GeoBIM²⁰ is widely acknowledged as an open initiative that addresses the data integration issues of both systems from specific use cases and from leading experiences. Furthermore, the new version 3.0 of CityGML brings new opportunities by refining constructions and building modules and creating new feature types to facilitate direct mapping from BIM data sets onto CityGML. Moreover, the introduction of the new space concept that already exists in BIM model can enhance

²⁰ <https://3d.bk.tudelft.nl/projects/eurosd-geobim/>

conversion between the two systems. All these improvements contribute to implementing DTs for cities and reducing data integration and interoperability issues while providing a convergence of the two worlds and connecting them in an efficient and interoperable way. Up to date, there are no official standards that are established to handle DTs for cities from a technical point of view. However, we should take advantage of the existing standards by ensuring convergence between them in a meaningful way with respect to the differences. In this work, the BIM models are used to typically update the semantic 3D city models and to increase their LoDs.

These standards are useful to build DT for cities, but it is a restrictive view since DTs for cities are not limited to static modelling but also take into consideration the dynamic objects properties and the city objects relationships. Thus, other standards are deployed, for example, IoT and sensor standards, web standards, OGC standards such as SensorThings API, 3D tiles, and I3S to name few. The use of standards helps to overcome the cost of data integration from various sources and increases usability. For example, in Germany, standardization efforts for DTs for cities are taking place to specify the main components of UDTs for municipalities. The consortium for DIN SPEC 91607²¹, "Digital TWIN for Cities and Municipalities," has been established in the framework of a joint project²² between the cities of Hamburg, Leipzig and Munchen, known as "Connected Urban Twins". The project's aim is to define common standards that identify the technical guidelines and non-technical aspects required to create DTs for cities.

Regarding the defined levels of data integration, for example, in the context of a DT project for a city, it appears that roads are required and the DT as it is initially designed is not considering this kind of city object. So, how to integrate or feed the DT when you have a specific requirement? And which level will be the most efficient to integrate this data? Three scenarios are possible. We can modify the DT conceptual schema model and add the required schema. Or, if we assume that the conceptual schema of the DT supports the theme road and we would like to update an attribute from an external database, in such a case, level 2 is the most appropriate level. Supposing that the roads are available through an intranet solution or stored in an administration database, in this case, the city objects will be visualized at the level of the viewer. In this sense, we are talking about level 3.

From a geospatial point of view, BIM and GIS integration is another example that could fit the different levels of data integration proposed in the scope of this work. Indeed, to update the DT for cities, the integration of BIM and GIS is relevant. To achieve integration between these two models, different approaches are feasible. Some examples use the Application Domain Extensions (ADEs) to extend the CityGML standard, allowing additional features to be mapped from IFC. Such an integration corresponds to level 1 integration. Level 2 is when a connection is established between classes in CityGML and IFC, including mapping of the elements and attributes. For level 3, CityEngine is one of the software that allows exporting 3D city models and BIM models, but without any semantic mapping. In this sense, the integration is performed at the application level, and nothing remains when the application is switched off.

Applying the different levels of data integration explained above, we classified DT for city initiatives according to their use cases. We have excluded some cities from this

²¹<https://www.din.de/de/forschung-und-innovation/themen/smart-cities/aktuelles/der-digitale-zwilling-fuer-staedte-und-kommunen-kommt--859000>

²² <https://www.connectedurbantwins.de/>

classification because currently no integration of dynamic data or simulation results is made and there is no available information about how the process of data integration is performed. **Table 2.3** presents the results.

Table 2.3. Data integration approach-based use cases and platforms.

CDT	Data integration approach		
	MCD - Level 1	BD - Level 2	Application - Level 3
Helsinki 3D	Energy ADE, Energy demand Heating demand and CO2 emissions Solar potential		Wind simulations
Espoo	Generic objects in CityGML	Generic attributes	
Rotterdam 3D	Energy demand	IoT and sensors data	BIM models
Zurich 3-4D			Urban planning, construction projects, street space in 3D via intranet solution, trees, archaeological objects, underground, power lines, BIM model, bridges. historical 3D spatial data, forests, open data catalogue
Virtual Singapore			3DEXPERIENCity
Digital Twin Munich	a conceptual model, combining a system dynamics model with a semantic 3DCM through a bidirectional data flow between the two models. Thus, an automatic change of the 3DCM is implemented for an urban densification use case based on the system dynamics model outputs.		
Rennes 3D			3DEXPERIENCity, subway planning and construction in Maurepas District of Rennes on traffic lights, bus station crowds, commuting times and neighborhood buildings, measurements from sensors for energy usage, traffic, pollution, and other metrics for early detection of anomalies
Virtual Gothenburg			Unreal Engine (simulations of traffic data), City Engine (parameters and attributes)
Herrenberg			COVISE- sensor network data-air flow simulation-urban mobility simulation-3D city model
Sofia- Bulgaria	Integration of address information, intersection of buildings and terrain		Computational pedestrian wind comfort- Wind flow simulations a digital twin using

			Computational Fluid Dynamics (CFD) Indoor air quality and thermal comfort and thermal comfort using CFD
DUET			3D model and urban data (IoT data, Geodata services), decoupled services that interact using central data broker. FIWARE smart data models Trusted urban data repository of city information
Fishermans Bend Digital Twin			Developing modules to convert different data formats and integrate, georeferenced and visualize.
NSW Spatial Digital Twin			Open data catalogue Data federation approach
Digital Twin Victoria			Accessing the data is done directly from its custodian (state government and agencies) Open framework of geospatial data services TerriaJS software. The platform does not store data. Instead, it connects to the data service via the platform.
Liveable City Digital Twin	Generic City Object class, for sensors data (stored the physical location of the sensors)	Generic attributes for several themes (buildings, roads and railways, water bodies) Update of the height values of the sensors	shadow analysis heat map of the trees visualization interface for the IoT data feeds directly via API
NUS-FRS	Ifc2CityGML		Integration of 3D campus, real-time data from sensors, energy demand simulation results from the City Energy Analyst tool, and occupancy rates from WiFi data.
EnSysLE project			Connection between the 3D city models and OGC web services to optimize the performance of the 3D urban energy visualization platform

The results show that different integration methods can be used for the same initiative. This depends strongly on the use cases. For example, the initiatives that use the semantic 3D city models integrate the data in similar ways. In fact, for the simulation results, level 1 and 2 are widely used by extending the schema model through ADEs and updating the attributes. Furthermore, most initiatives implement the DT by creating a 3DCM using the CityGML 2.0 standard. Hence, data integration is primarily explored through the CityGML schema, which enables the extension and interoperability of 3D models across various applications. Future developments for several initiatives consider including additional objects such as underground infrastructures and open spaces to name a few. These implementations will probably use the generic objects and attributes of the CityGML

standard. It is also important to consider that levels 1 and 2 are usually interrelated. Indeed, if we extend the conceptual schema, we will necessarily need to feed the classes with the required information. Level 3 (application-based data integration) is also used in different initiatives, particularly the Australian initiatives that are built in similar ways, which explains their interest in building a National Digital Twin. The application-based approach is the most encountered kind of platform since it can be challenging to take into consideration a general and unified conceptual model in the conception phase of the creation of DT for cities.

To conclude, we can state that the integration of data in a DT for cities is one of the main issues that need to be tackled (different sources, semantics and structures). For this reason, the integration needs to be designed according to the level of maturity. Furthermore, data integration is guided by the type of data, the simulation and the application requirements. In the DT initiatives, there is usually a need for visualizations that consider multiple end-users. Furthermore, the DT initiatives currently implemented in practice tend to focus on the development of DT-based use cases. This explains the different levels of data integration deriving from the scope of this work. In the future, further integration levels can be conceived, or we can design a generic DT that includes the relevant city objects, services, or processes, and on the basis of which DTs applications can be performed later.

2.6 Discussion of findings

The results obtained from our scientific literature review and survey analysis enhance other related works that address the challenges of DTs for cities, namely the lack of a common definition and data integration. Indeed, our findings give an in-depth analysis of how the DTs are defined and implemented in practice, which helps us understand the current realizations and document them. By clarifying the DT's definitions and extracting the key features and components, we can easily relate what the maturity level of the 3D digital model is. Thus, conducting a systematic review of DTs with related modeling concepts and putting them into perspective to define their convergences and divergences. Reviewing related works used in this paper, our findings extended the study established by (Deprêtre et al., 2022) that pinpoints the conceptual and technical aspects of both DTs and CIM. In addition, a review conducted by (Ketzler et al., 2020) examines historically the DT definitions and identifies similarities and possible variations with 3D city models. However, currently, there is no study that compares and discusses, from a theoretical and practical point of view, DTs with 3DCM and CIM as modeling approaches. Furthermore, no related work considers SDI as an approach that can be applied in the given context where a DT consists of a system of systems at multiple levels of scale and complexity. One of the key findings is that it is important to clearly define the concepts and approaches, thus understanding the input layer of DTs for cities from practice. Indeed, current implementations confuse the concept of DTs with other notions and sometimes consider it as a technological evolution of what has been achieved in 3D urban modeling.

Through listing the current DTs definitions from the literature review, we intend to study the differences and extract the common features that characterize the DTs. The aim of this section is not to provide a complete and unique definition of DTs for cities but rather to identify and discuss in length the various definition tracks reviewed in this work. Hence, considering DTs for cities as systems of systems is the most highlighted track, e.g., 15 papers have defined DTs for cities as a digital representation of the city elements that

function as an ecosystem of twin systems without narrowing the scope to the 3D components. The analysis of the definitions reflects the key features and main components of DTs for cities. We primarily focus on the heterogeneity of the data since DTs for cities can be conceived as data containers. In addition, the data link between the virtual and real worlds and the data integration on unique and generic platforms are highlighted as the most questionable characteristics of DTs that need to be addressed in future work.

Following the systematic literature review of definitions and concepts, the DTs for cities terminology is also a fundamental challenge that needs to be addressed at an early stage. Especially, DTs for cities enhance the general collaboration between different stakeholders who are from various backgrounds. Such terminology ambiguity may lead to different expectations as for the DTs definitions. Hence, our findings indicate that there are several terms deployed in practice to label DTs as cities. Thus, Urban Digital Twins and City Digital Twins are the most common labels between academic and practical fields. Another consideration that is identified in the scope of this work, while analyzing the three parallel tracks, is that DTs for cities can be misused not only in a terminological sense but also as regards other concepts and approaches. Therefore, we discuss the overlaps and the differences between DTs for cities and related approaches, namely 3DCM, CIM and SDI. Our findings indicate that the 3DCM is a good starting point for creating DTs for cities since historically, the literature review used 3DCMs to perform various urban applications. However, continuous enrichment of the semantic 3DCMs is mandatory to meet the DT's specifications. On the other hand, when comparing DTs for cities with CIM, we intend to identify which concept belongs to the other since CIM and DT are under the same umbrella (similar conceptual and technical realities). However, the review analysis shows that the CIM is mainly considered as the input layer of DTs for cities. Another reflection that gains perspective in the context of DTs for cities is the setting of SDIs as the cornerstone of creating a solid geospatial DT. Indeed, SDI will facilitate seamless access to the data hub as well as efficient and standardized data integration. The results from the comparative analysis show that the state of the art is more developed working with the 3D city models and CIM than the SDIs, only a few papers identify the necessity of having an SDI in the context of geospatial DTs.

Regarding the survey results, the findings are in line with the most frequently discussed issues regarding the definition of the concepts and the main approaches revealed in the systematic literature review. Furthermore, the exploratory survey depends highly on the results of the literature review since almost all questions are derived from it. Thus, the literature review highly facilitates the interpretation of the results, especially since 11 responders are from academia and DTs for urban settings are one of their research hotspots. Our first contribution from the systematic review and survey analysis is to enhance the existing related works, mainly the work of (Ellul et al., 2022; Lei et al., 2023), for which authors identified and discussed the technical and non-technical challenges that hinder the full implementation of the UDT. Our study supports their results by providing a more in-depth analysis of the approaches related to the DTs rather than merely focusing on defining the concept and highlighting the challenges.

The second major contribution of this paper is to get an overview of some ongoing projects and analyze the technical DTs implementations according to 9 criteria. The list of current realizations is not exhaustive, since we do not intend to document all the initiatives, however, we aim to identify the technical requirements to create DTs for cities. Such findings enable us to extend the work conducted by (Lei et al., 2022) to assess and benchmark 3D city models according to 47 criteria. Similar reflection is applied in the

context of the DTs initiatives, as we are far from having a single shape and application of DTs for cities. The results allow us to bridge the gap between the theoretical findings and the technical perspectives and gain insights from both fields.

While the work provides comprehensive insights, we attempt to further investigate these initiatives. For this, we first compare the different initiatives according to the maturity level of integration with respect to the nomenclature that defines a digital model, a digital shadow and a digital twin. According to the results collected, all the initiatives deploy the term DTs in their projects; however, the current implementation does not really meet the conceptual definition of DTs, which consists in ensuring the automatic connection between the real world and the virtual copy.

Then to address the level and the approach of data integration, we focus from a technical perspective on how the data are integrated in DTs for cities. To reach the third contribution of this work, we analyze the DTs initiatives according to the use cases. Since, the data are incorporated in DTs for cities according to the application and the data type. From a general standpoint and based on the initiatives discussed in this paper, the application-based approach is the most popular implementation in the context of DTs, it consists of a decentralized approach where accessing the data is done directly from its repositories and integrated at the viewer level. The initiatives focused on level 1 and 2 can be challenging as defining a conceptual model and managing urban databases are complicated by the fact that all city elements and their interdependencies must be included, modelled and stored in an efficient way. This remains the objective of creating a generic DT. However, this must be done fully independent of technologies, systems, standards and software. Which cannot be applied to the practice that uses existing data models and standards and works on fostering the data interoperability between different systems.

2.7 Urban Digital Twin Definition Adopted in This Thesis

Based on the analysis presented in this chapter, a preliminary definition of Urban Digital Twins is adopted here and refined in later chapters. An Urban Digital Twin is a digital representation of the city and its subsystems, represented by semantic urban models and connected to heterogeneous real-world data in order to support integration, analysis, visualization, and lifecycle-based urban applications. It is built upon semantic 3D city models enriched with multisource data, often including dynamic or near-real-time information, so as to enable data flows between the physical and virtual worlds. Its role is not limited to city visualization, but extends to monitoring, simulation, prediction, and decision-making through an interoperable and integrated framework.

In this thesis, the defining feature of an Urban Digital Twin is therefore not a strict one-to-one replication of the physical city, but rather its capacity to structure, integrate, and harmonize heterogeneous urban data within a coherent and integrative representation across scales, domains, and lifecycle stages.

This definition highlights data integration and interoperability at the core of Urban Digital Twins and positions them as evolving systems capable of supporting both static and dynamic representations of urban environments.

2.8 Conclusion

Digital Twins have captured high interest in the urban and geospatial domains at a city-scale level. They face many challenges that hinder the full deployment of the concept, namely the lack of a common definition, which leads to several frameworks and implementations in practice. Some research has defined DT for cities, which generates multiple, and sometimes very broad definitions. Such definition diversity leads to ambiguities among practitioners, which motivated us to conduct this work to bridge the gap between the concepts and current realizations of DT for cities. Our findings are based on the review of literature and an in-depth analysis of the results collected from the online survey. According to scientific literature, we refine the DT for city definitions and extract the common features. The survey results were studied as well to measure how closely the practice matches the theoretical and conceptual definitions. A comparative analysis with related concepts was also conducted as DT was misused with other related terms in the literature (i.e., 3D City model, City Information Model and Spatial Data Infrastructure). A comprehensive study of the DT initiatives is provided according to nine categories that give various stakeholders the technical requirements to build a DT. This analysis of DT projects will assist future initiatives in learning from lessons and experiences gained in the framework of the projects' development. This is a synthetic repository of DT initiatives acquired from various articles and websites of different projects. These results will facilitate the understanding of DT's current implementations. Further analysis is provided in this work regarding the maturity of different levels of integration. Thus, the initiatives are classified according to the data connection between the digital and virtual worlds. Furthermore, a new classification method to assess the DT implementations is given based on data integration methods. The aim is to understand at what level the data integration is performed. Accordingly, three levels are suggested: Level 1: conceptual data model; Level 2: database level; and Level 3: application level.

The data integration assessment led to the conclusion that the differences among DTs implementations are significant. Each initiative has a specific framework to incorporate their data according to the city requirements.

This study provides an initial step toward bridging the gap between the conceptual definitions and the technical development combining the results from the literature review and from the modest insights provided by international experts. However, some limitations should be revealed in the future work. Regarding the literature review, 26 articles are collected to study the DTs for cities definitions. Most of the documents are conference papers rather than review articles. This confirms the early stage of the DTs concepts. Another limitation is to properly measure how the literature findings are faithful to the practice. Although, current implementations of DTs highlight the general interest given to the topic among urban and geospatial communities, however, the survey is based on few participants. Most of the responders are from research community, only 2 from industry and one from city government share with us their thoughts regarding the DTs for cities and give us insights about their work. The number of survey responses could be much higher to get valuable interpretations and generating correct insights. More input from industry and city government will certainly increase the findings and enhance the synergy and the collaboration between literature and practice. Finally, our work does not provide a complete list of the DTs initiatives but rather, discussed them considering 9

comprehensive criteria that are relevant from a technical perspective. However, there are various criteria that should also be covered in future work, namely the organizational and the social aspects.

To conclude, it is important to state that data integration is ubiquitous in the overall technological framework of building DT for cities, from data acquisition to data updates. Addressing the data integration issue will help cities achieve more mature versions of DTs. Indeed, plenty of data is available (i.e., IoT data, simulation results, prediction results, external databases, BIM models, system dynamic models, etc.) with different formats, data quality, and level of details. Each type of data will probably have an effective level of integration.

For future perspective, this data integration classification can be projected to any DT project, especially since our questionnaire will remain open during our research to understand the technical requirements of implementing DTs in practice. However, the main question to answer is: *at what level is it appropriate to integrate?* In our future work, we intend to study the different levels of data integration as explained in this work. We believe that addressing the data integration issues and analyzing the effectiveness of the different levels will provide better understanding of the DTs for cities implementations with respect to the data heterogeneity and the complexity of the urban environment.

CHAPTER 3: Data

Integration across UDT

Lifecycle:

Conceptualization

Preface

Following the clarification of concepts and the identification of key components of Urban Digital Twins (UDTs), this chapter addresses one of their main technical challenges: data integration. While UDT concept is increasingly promoted as the new paradigm to address contemporary urban issues ranging from climate change to mobility and energy transition. Their effectiveness largely depends on their ability to handle heterogeneous data and models in a consistent and standardized manner. In practice, interoperability and data integration remain among the most critical issues encountered when evolving semantic 3D city models toward urban digital twins.

Although the importance of data integration is widely acknowledged in both research and practice, the concept itself is often highlighted, but rarely articulated in a structured and systematic way. The aim of this chapter is to address the following research question:

RQ2. How is data integration addressed within the emerging concept of UDTs?

To this end, the chapter analyzes how data integration is performed in UDTs, how it varies across current implementations, and how it can be framed within a coherent conceptual framework. Existing UDT implementations reveal a lack of both specific and generic integration approaches. Indeed, integration mechanisms differ significantly depending on the use case and the types of data involved (e.g., IoT streams, simulation outputs, external databases, BIM). While this diversity reflects the growing maturity within the field, it also introduces ambiguity. Indeed, despite the widespread use of data integration as necessary step in each UDT implementation, there is no shared understanding of how data should be integrated, at which level, or at which stage of a UDT lifecycle.

Similar integration and interoperability challenges have long been encountered in related geospatial domains. However, Urban Digital Twins introduce additional layers of complexity due to the scale, heterogeneity, and dynamic nature of urban data infrastructures, as well as the need to support continuous interaction between data, models, and applications.

To address this gap, this chapter proposes and formalizes the concept of data integration levels for UDTs. Three main levels are defined and summarized in **Figure 3.1**. **Level 1**, the conceptual data model level, focuses on extending or modifying the underlying data model to incorporate new data. **Level 2**, the database level, involves transforming and adapting data to match model requirements and populate specific attributes or classes. **Level 3**, the front-end or architecture level, relies on customized architectures to integrate, link, and visualize data through web services, APIs, and viewers.

This chapter builds on the article “*Data Integration across the Urban Digital Twin Lifecycle: A Comprehensive Review of Current Initiatives*” (Jeddoub et al., 2024), which analyzes data integration levels in relation to the UDT lifecycle. By anchoring the integration discussion to lifecycle stages, the chapter addresses fundamental questions namely:

RQ3. How can data integration be defined and formalized according to different approaches across the UDT lifecycle (creation, use, and update)?

Through a comparative analysis of existing city-scale implementations, the chapter illustrates the practical application of the proposed integration levels, highlights their respective advantages and limitations, and discusses how different configurations may be appropriate depending on use cases and data heterogeneity.

Rather than seeking a bespoke data integration approach for UDTs, this chapter adopts a pragmatic perspective by introducing integration levels as an analytical framework. These levels aim to structure existing practices by distinguishing where and how data integration occurs within a UDT, while acknowledging that implementations may adopt different approaches. Current initiatives tend to prioritize the creation and use phases of the lifecycle, while relatively few demonstrate robust mechanisms for data maintenance and update integration, revealing a critical gap in the UDT lifecycle.

By formalizing data integration through both levels and lifecycle stages, the proposed LoI framework does not attempt to prescribe a one-size-fits-all solution. Instead, it offers a structured, domain-independent perspective for understanding, comparing, and guiding integration approaches. This chapter therefore provides conceptual insights for both researchers and practitioners to reason about interoperability, support integration, and anticipate challenges inherent in evolving UDT ecosystems. As cities increasingly invest in Urban Digital Twins, achieving maturity in data integration is fundamental for moving beyond isolated demonstrations toward scalable, interoperable, and sustainable urban digital infrastructures.

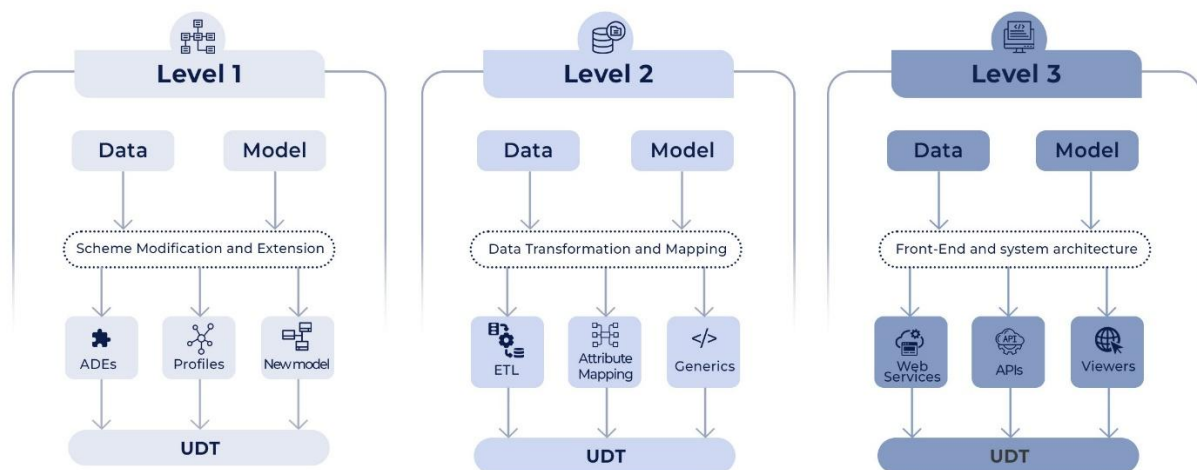


Figure 3.1. Conceptualization of Levels of Integration.

Based on the article (Jeddoub et al. 2024)

Data Integration across Urban Digital Twin Lifecycle: A Comprehensive Review of Current Initiatives

Jeddoub, I., Nys, G.-A., Hajji, R., & Billen, R. *Ann GIS* 2025; 31: 367–386.
doi:10.1080/19475683.2024.2416135 (2024).

Abstract: Challenges related to data integration and interoperability were raised recently under the auspice of the Urban Digital Twin (UDT). This new paradigm shows its potential to address current city challenges. However, to maximize its outcomes at the city scale, we should tackle the fundamental issues related to data integration. Indeed, various Digital Twin (DT) frameworks have been developed in practice. Based on these implementations, we define three main levels of data integration. The first level involves the extension of the data model to handle new information. The second level supports data by default, and the data needs to be transformed to meet the model requirements. The third level performs the integration at the front-end level with the help of system architecture. The aim of this work is to analyze, illustrate, and guide the effectiveness of different data integration approaches. This chapter unpacks the levels of integration according to the corresponding UDT lifecycle phases (i.e., creation, use, and update phases). It highlights the challenges and potentialities of data integration levels and offers DT designer conceptual guidelines related to data integration. Furthermore, current and theoretical data integration scenarios are extracted and investigated, considering several types and sources of data. This research provides a comprehensive analytical framework for data integration within UDTs, where some of the current operational UDT are examined based on the various integration levels of life cycle data. While the state-of-the-art identifies data integration as a major challenge for the full implementation of UDT, it is not explored in depth, and the integration is only addressed from a case study-specific perspective, according to the data availability and the UDT requirement. Hence, this framework provides a generic and application-independent overview of the different levels of data integration based on the UDT lifecycle inspired by the Spatial Data Infrastructure lifecycle. This article provides first conceptual insights of data integration levels to create, use, and update UDT. Furthermore, the current emphasis is on the creation and use phases of the lifecycle, which lacks a concrete illustration of the update phase. Indeed, it limits the practicability of the data integration levels in the maintenance phase.

Keywords: Urban Digital Twins; data integration; level of integration; data interoperability, DT lifecycle.

3.1 Introduction

Urban Digital Twins (UDTs) are fast establishing a foothold as a contemporary trend and a research hotspot worldwide in the urban and geospatial fields. They have triggered a huge interest in the current technological discourse. Nevertheless, this new technology comes with its own challenges, restricting its full implementation at the city level. For instance, data integration has significantly triggered considerable interest as one of the severe challenges while moving towards the implementation of UDTs. Certainly, integrating the existing data might address the incompleteness of the datasets, minimize the costs of new acquisitions, and create new and insightful data flows towards more advanced use cases and urban applications. However, the reuse and integration of available and accessible data raise data interoperability and integration issues (Lei et al., 2023; Noardo, 2022). Furthermore, recent studies leverage novel data integration paradigms and challenges of Generative AI models and Urban Digital Twins under the Cognitive Digital Twin concept (Xu et al., 2024). This AI-driven Digital Twin is an emerging form of the traditional DT that is augmented with cognitive functions and extended with semantic technologies. That means that Cognitive Digital Twins is a sophisticated and enhanced stage of DT implementation. They reveal a promising evolution of the current DT concept by enhancing DT capabilities towards more intelligent, autonomous, self-learning, and reasoning twins. This enables the digital counterpart to learn from the provided data to address the data complexity based on the deep learning approaches, simulate and predict complex scenarios, and improve autonomous and real-time decision-making. The particularity of this cognitive digital twin is the capacity to continuously evolve according to the physical world across the entire lifecycle (Zheng et al., 2022).

To highlight the close relationship between data integration and UDT implementation, it is first important to clarify how Digital Twins (DTs) are conceived within the scope of this work. DTs are defined as digital models of the real world, allowing data exchange between these two counterparts. Since the concept has been used interchangeably in a transdisciplinary way according to different backgrounds, we adopt the following definition from an urban and geospatial perspective: the urban or geospatial DTs use 3D city models (3DCM) enriched with semantic information (i.e., enhancing the potential of geospatial data), often coupled with near real-time data (emphasizing the dynamic aspect of DTs), enabling data flow between the real and virtual worlds, offering various sets of analysis through simulations, predictions, and visualization tools (web-based applications, analytical tools or game engines platforms), and creating a one-stop platform (i.e., a multiscale and multitemporal database) of various datasets and models enabling them to investigate and address city current and future social, economic, and environmental challenges, to name a few (P. Alva et al., 2022; Jeddoub et al., 2023; Ketzler et al., 2020; Stoter et al., 2021).

Following the mapping of the challenges with the DT lifecycle, data integration is commonly considered in the processing and generating phases (Lei et al., 2023). However, the use of DTs, more precisely, the use of 3DCM (as integral components of UDTs) and as input data for simulations and urban analysis generates, in turn, new and specific datasets according to the use case that need to be integrated into UDTs. In addition, the maintenance phase (i.e., the update) requires updated data as it becomes available to be ingested and processed. This implies that integration is part of not only the processing

and generation phases of the DT lifecycle but also extends to the stages of use, maintenance, and updating.

Current implementations reveal a lack of a specific and generic approach to data integration. This is justified by the fact that, on the one hand, data integration is carried out through various approaches, as will be discussed in the following chapter. On the other hand, for each specific case, the data to be processed varies, and so does the integration approach. Moreover, this integration issue has always been encountered in related and similar concepts (3DCM, spatial data infrastructure, city information models, etc.) for more than a decade, yet there are still considerable efforts being conducted to achieve seamless integration and enhance data interoperability. Thus, data integration needs to be tackled in a comprehensive way, especially while using 3DCMs as the integration basis for UDTs. Using standardized models (i.e., CityGML as the common data model for semantic 3DCMs and recently UDTs) shows its potential to solve some integration issues. For instance, the enrichment and integration of external data into the 3DCMs are feasible through application domain extensions (ADEs) according to the use case. However, as many cities move toward the implementation of UDTs, integration of different data sets (i.e., Internet of Things (IoT) data, simulation outputs, external databases, BIM, etc.) with different formats, sources, and data quality is still considered one of the first issues in practice.

To support data integration, we need to understand how data might be integrated into UDTs in practice according to the corresponding DT lifecycle stages. This lifecycle, which was borrowed from the spatial data infrastructure stages, mainly consists of three main phases: **creation, use, and maintenance**. The purpose of establishing such a link between data integration and the different lifecycle phases is to provide fundamental insights on the underlying questions: *How do we incorporate data to create a DT, which may be a model or a system of systems? How do we embed data resulting from the use phase into the DT? And finally, how can we guide the data integration process when it comes to updating the DT?*

The main contribution of this chapter is to analyze, illustrate, and guide the effectiveness of different data integration approaches. The chapter explores the various data integration levels during the creation, use, and updating phases of UDTs, considering real-world implementations at the urban scale (namely city and district scale). This chapter mainly offers a comprehensive analysis of data integration levels throughout the UDT lifecycle, comparing their advantages and limitations and illustrating them with current operational implementations. However, the analysis presented in this chapter has limitations, notably in terms of the scope and selection of the Urban Digital Twin (UDT) projects considered. For instance, we are unable to document all the initiatives since several initiatives are using the term digital twin, whereas in reality this is simply a digital model that fails to reflect the full UDT functionality and features. Thus, a series of criteria were applied to filter the different initiatives included in this study, in line with the levels of integration across the lifecycle. First, the initiative's scale is the city, precinct, or district scale. Second, initiatives need to clearly emphasize the geospatial component and have geospatial data, notably the 3D city model, which is a key component in the creation of a UDT. In addition, the UDT project is operational and integrates a variety of data based on the use case requirements (i.e., air quality data, energy data, social data, to name a few). Finally, the UDT maturity model is also taking into consideration (e.g., based on the

CITYSTEPS maturity model with an emphasis on stages from 3D static stage to real-time synchronization and autonomous implementation stage (Haraguchi et al., 2024)).

While UDT has many inherent technical challenges, our focus is on data integration within UDT. Our choice to address this major UDT technical challenge is that integration and interoperability are still considered the first issues in practice, especially that the complexity of data has increased under the umbrella of UDT as various multi-source data and models are integrated. In this regard, the remaining chapter is structured as follows: Section 3.2 defines data integration and reviews related issues in 3D GIS and UDTs. Section 3.3 explains the levels of data integration. Section 3.4 investigates the challenges and potentialities of data integration levels. Section 3.5 represents and analyzes the levels of integration into the lifecycle of DTs. The same section exemplifies the different levels according to some ongoing implementations in practice. Section 3.6 highlights the main findings and gives a data integration overview. Section 3.7 concludes this work and gives a glimpse of future perspectives. These sections will enable the UDT developers and users to clearly understand the fundamental approaches of integration that rely on clear semantics and specific requirements and to prove the relevance of data integration levels based on different scenarios showcasing their applicability through some academic implementations.

3.2 Data integration and related issues

Many urban applications focus on data integration as an intrinsic phase in 3D city modeling (Stoter et al., 2020). Although the challenges related to multisource data integration have been recently tackled in different studies with a strong focus from the geospatial community, the challenges of data integration evolve increasingly under the auspice of the UDTs. Historically, spatial data infrastructure (SDI) has been developed at different levels and scales to assist effective data integration, usage, and sharing among different policies addressing technical and non-technical integration issues (Hu, 2017; Janowicz et al., 2010; Kotsev et al., 2020). However, the actual SDIs (i.e., INSPIRE and the Open Geospatial Consortium "OGC" catalogs) did not yet meet the maturity required for managing UDTs data integration challenges. Recent studies have suggested incorporating SDI within UDTs. (Knezevic et al., 2022) proposed the implementation of an extended catalog system in Germany, leveraging metadata and the open-source catalog CKAN²³ software package. Another related work established a UDT and explored the development of an SDI-based energy domain using OGC standards to manage urban energy building data (Santhanavanich et al., 2022).

Furthermore, (Noardo, 2022) proposed a detailed and practical workflow for suitable multisource data integration, starting from defining the requirements based on the use case to the update and writing of metadata. The overall workflow can serve UDT implementations as it defines guidelines to properly understand and handle the fundamental integration issues based on the data coming from practice. For instance, the possibility to use data originally acquired for a specific use case in another application and vice versa is one of the benefits that UDTs have brought forward by focusing on integrating multiple domains.

Before we outline the main challenges of data integration, let us clarify the concept of data integration and specify what it covers. Indeed, data integration is acknowledged as the

²³ <https://github.com/tum-gis/SDDI-CKAN-Docker?tab=readme-ov-file>

process of gathering and homogenizing multiple datasets from different sources depicting the same physical world feature and consolidating them into a unified and consistent format (Abdalla, 2016). However, this definition is much more related to data fusion, merging, and harmonization through the encoding format and common database schemas (Li et al., 2020; Weil et al., 2023). These phases are regarded as components of data integration. Some relevant methods are available in the literature to merge and harmonize the data from various sources. Nevertheless, effective integration is partially achieved due to the lack of metadata, the differences in spatial and temporal scale, the geometry mismatch, and the lack of clear semantics, as many stakeholders are involved while moving towards the implementation of UDTs. Indeed, rich metadata should guide data integration, along with the definition of the use case and application requirements. Moreover, data integration is interlinked with the modeling paradigms, the storage, and the intended use. In this regard, data integration is conceived as the process of combining, transforming, and connecting multi-source datasets into a single shared data model, database model, or one-stop platform, ensuring an accurate and insightful representation of the real world. It also takes into consideration, in the urban and geospatial scope, the enrichment process to create a complete and detailed 3DCM, enhancing further urban applications.

The integration within the UDT scope is particularly complex due to the lack of a common technical framework for data integration. On top of that, the integration focus is different when considering the lifecycle of DTs. Some authors focus on the data integration in the creation phase, which basically starts with the creation of 3DCMs where geospatial urban features need to be incorporated (Dimitrov and Petrova-Antonova, 2021; Fricke et al., 2023; Khawte et al., 2022). While other authors focus on the integration of specific data sets that could be dynamic, i.e., near real-time data from sensors and IoT data, or simulation data that, in turn, are integrated into the 3DCMs for further urban analysis (Chaturvedi, 2021; Santhanavanich and Coors, 2021).

Another data integration focus that gains interest while creating and updating UDTs is the integration of BIM and geoinformation, known as GeoBIM (F. Noardo et al., 2020).

When integrating various datasets, a considerable number of challenges need to be addressed, namely, syntactical, structural, and semantic levels of interoperability (City information modelling and urban digital twins | IEC, 2021). The Open Geospatial Consortium (OGC) has formulated international standards and data models to tackle interoperability and, therefore, data integration issues (i.e., CityGML/CityJSON, SensorThings API (STA), and OGC API features, to name a few). Thus, open standardized data models and exchange formats are required to enhance data interoperability and integration. Both (i.e., integration and interoperability) are technical barriers towards 3D city modeling and, more recently, UDTs (Billen et al., 2014; Döllner et al., 2006.; Kolbe et al., 2020; Kolbe and Donaubaue, 2021). These challenges are interrelated. This means that data interoperability is a requirement as well as a part of data integration. Although data integration aims to foster usability and tackle the incompleteness of the datasets, data quality, availability, and conversion are serious challenges that hinder effective data integration. On the other hand, fundamental issues (such as semantic heterogeneities and information loss, limited or lack of metadata, geometry issues, standards inconsistencies, etc.) still require consideration while moving towards the implementation of UDTs. Further, understanding the semantic differences between each standard and each domain is vital to breaking down the data silos and reducing the complexity of existing standards. In short, creating complex and connected systems such as UDTs requires integrating various datasets from different domains. Indeed, effective integration of various datasets

is a key technical driver. However, the main question that should be answered is: **what are the data integration levels, and what is the appropriate level of data integration based on the UDT lifecycle?**

3.3 Conceptualization of data integration levels for UDTs

The heterogeneity of the technological framework and architecture to implement UDTs is as diverse as the data available in practice. However, while creating UDTs, cities reuse the existing data and generate new data sets through data integration and enrichment.

The investigation of DTs implementations from practice deployed different levels of data integration based on the use case requirements (UDTs-based use case). Furthermore, there is no universal definition of what UDTs represent and few related works define the minimum technical requirements to develop a true "UDT". Thus, understanding the data integration approaches is fundamental since these levels in their turn pose their own challenges.

Based on the literature and the classification of a significant number of UDT implementations, three levels of data integration were defined: the conceptual data model (CDM), the database level (DB), and the front-end level (see **Figure 3.2**).

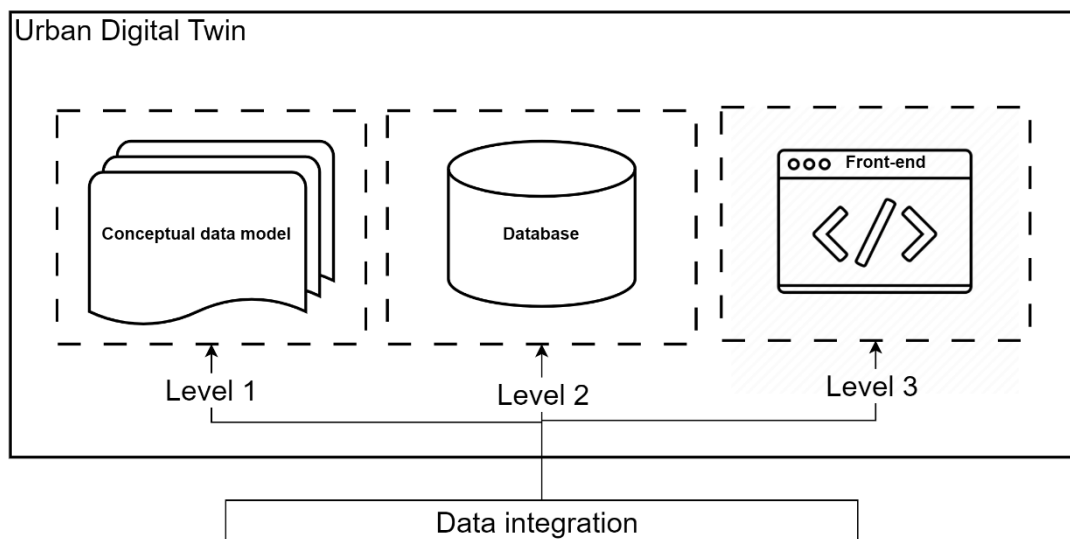


Figure 3.2. Levels of data integration: Possible ways for integrating data into UDT.

3.3.1 Conceptual data model level (CDM)

The **conceptual data model (CDM)** is defined as the schema that structures the data and the relations between entities and attributes.

In contrast, from a technical perspective, existing data models and standards are deployed; either they are extended to cover new themes and new features, or multiple data models or subsets of data models are merged based on a new unified data model based on the DT requirements. Thus, performing data integration at the conceptual data model is the upper level of integration.

At the time of writing, there is no official conceptual data model designed to cover all UDT requirements. This offers the user the possibility to modify and extend the existing conceptual core model to integrate objects that are not already represented by any other class. Indeed, integrating at this level implies adapting the core model using extensions provided by the current CityGML standard and its encodings (JSON and 3DCityDB). In fact, CityGML is the most used data model for semantic 3DCMs. Its conceptual data model has been considered a data hub and integrative platform for urban and geospatial data for decades, allowing data integration and the creation of a comprehensive Urban Information Model (UIM). They are therefore considered the best candidates for introducing level 1 in the context of UDTs serving as an abstraction of the physical world and an integration basis. This data model is generally extended to cover and incorporate different data for generic purposes (i.e., adapting the conceptual data model to a national or regional context) or to support specific use cases and applications through ADEs. This level offers a direct feed into the database.

3.3.2 Database level

The **database level** could be defined as the process of converting the data to be integrated to match the conceptual data model and the database schema. In fact, the CDM is not edited; otherwise, we will be talking about Level 1. The database level allows for feeding or updating specific classes or attributes without altering the semantic consistency of the data model. At this level, the data is adjusted for database feeding. ETL (Extract, Transform, Load) processes are deployed to transform the data to meet the data model structure and schema. For this purpose, the model is unchangeable, but rather the data needs to be adjusted in line with the data model.

3.3.3 Front-end level (client side)

This level represents the **UDT's front-end**. Integration at this level could be performed via various interfaces and tools (i.e., GIS tools, web applications, geoportals, dashboards, linked data interfaces and game engine applications, to name a few).

Web-based applications show their capabilities in visualizing and handling complex 3D data on modern browsers. While moving towards UDTs, 3D data integration on the Web has been flourishing. This data integration level represents the most common approach to multisource data integration in the current UDT implementation, as previously investigated in Jeddoub et al. (2023). It allows the data to be integrated into common interfaces and handle different data exchange formats, achieving syntactic interoperability. Furthermore, many cities implement a distributed architecture for data integration and management and disseminate the 3D content “as data-based services” where many stakeholders can interact while maintaining their own data. Hence, the data can be served and visualized on the web by various parties without following a data integration process at the data model level. In addition, web-based UDT platforms typically expose APIs that enable developers to access and integrate data efficiently, supported by widely adopted web-oriented standards.

The ambition towards fostering open data through the web and making it accessible to multiple stakeholders enhances the need to investigate the development of web UDT platforms. However, combining various data and models into a one-stop platform may

lead to some inconsistencies. For example, data coming from practice has diverse heterogeneities ranging from the semantic representation to the syntax, the structure, and the geometry. Furthermore, the data may encompass various spatial and temporal scales. However, from a technical perspective, current platforms based on WebGL JavaScript libraries, such as CesiumJS, increasingly support multi-format data integration. Indeed, Cesium, the common open-source platform, is used to integrate and visualize the required data. The platform supports several formats, and the data basically goes through a data conversion process. Nevertheless, integration might be affected at the semantic level. Indeed, both semantic and syntactic interoperability are extremely valuable while developing collaborative platforms such as UDTs. In parallel, Semantic Web approaches provide mechanisms for integrating heterogeneous built environment data through shared, ontology-driven representations, thereby addressing semantic heterogeneity and data silos (Chadzynski et al., 2021; Tan et al., 2025).

The data integration at the front-end level based on the web still represents some challenges in practice that are addressed within the geospatial community, namely through data standardization and interoperability. While web-based platforms have shown their capabilities in handling various 2D and 3D geospatial content, game engines, in turn, have attracted considerable attention as effective tools for UDTs. Although browser-based platforms such as CesiumJS and ArcGIS Maps SDK for JS facilitate user accessibility as well as enhancing comprehensive support of geospatial data, their visualization and interactive capabilities are still limited, especially when operating on large urban scales such as cities. This is where game engines have found room for development.

Game engines like Unreal Engine (UE) and Unity bring new opportunities to access and visualize real-world data represented through urban and geospatial data with high visual rendering in an immersive way. Furthermore, the game engine plugins available in practice, such as the UE plugins for Cesium Web Globe and Esri's Web Globe, consolidate the link between 3D geospatial technologies and game engine platforms. Nowadays, game engines are considered a trend for data integration, particularly under the umbrella of UDTs. From data standardization and interoperability perspectives, an early exploratory study was conducted to test the usability, compatibility, and viability of existing geospatial OGC standards, namely, 3D Tiles, I3S, and 3D GeoVolumes API, with game engines (i.e., UE and Unity) based on Cesium and ESRI's plugins (Würstle et al., 2022). The authors illustrate through three use cases the usability and applicability of the OGC standards in the framework of game engines. Furthermore, challenges faced with the implementation of the use-case-based OGC standards were documented. These limitations need to be studied in a comprehensive way since they affect the final model.

In conclusion, various architectures are proposed in practice and are mainly based on a web browser platform, namely Cesium and ESRI solutions such as ArcGIS online and JavaScript for ArcGIS. Cesium is widely used in academic UDT initiatives compared to ESRI tools. This is certainly due to their open-source initiative. The browser-based geospatial platforms show their capabilities in accessing, integrating, and visualizing 3D content. Although they have great support for geospatial OGC standards, they are still limited in terms of effective rendering and interactivity. Hence, there is a need to further explore the potentialities of game engines.

3.4 Data integration levels: Pros and cons

The data integration levels have their pros and cons when developing a UDT. This section discusses the strengths and limitations of each data integration level and proposes guidelines for users in terms of their effectiveness.

Table 3.1 highlights some of the advantages and disadvantages of the data integration levels. The following list of pros and cons is not exhaustive, and further improvements and drawbacks may emerge in the future.

Table 3.1. The pros and cons of data integration levels.

	Level 1	Level 2	Level 3
Pros	+ Semantic interoperability + Model extensibility + Coherent and harmonized data model + Schema validation + Data import “as is” (no conversion needed) + Direct load to database	+ Straightforward data model compatibility + Native support of city objects (i.e., CityGML urban objects) + Use of generics attributes (i.e., CityGML generics) + New versions of data model enhance direct data integration.	+ Syntactic interoperability + Separated and independent database management systems + On the fly simulations (based web services and processes) + No data conversion + Various data format support
Cons	- Model complexity - Compliance with the Application Domain formalities (i.e., CityGML ADEs) - Large data model (huge number of classes, attributes, properties) - Compatibility with new data model versions. - Software support for extensions is still limited.	- Preprocessing needed - Data loss (semantic loss) - Mapping between entities and attributes could be tricky.	- Heavier bandwidth. - Limited interactive client-side

According to **Table 3.1**, level 1, which performs integration at the data model, provides a range of advantages, including semantic interoperability and compatibility, taking advantage of the model's extensibility and scalability, and the possibility of schema validation after extending the data model. Furthermore, data can be imported directly without the need for a data conversion process. In addition, data is directly fed into the database, or more specifically, when the DT's technological framework relies on databases for data management and storage. While there are significant advantages offered by this level of integration, there are also a series of challenges that should still be addressed. Extending a data model certainly allows new classes, entities, and attributes to be managed, but this may increase the complexity of the data model (for example, adapting a CityGML model presents a range of challenges given its complexity). Besides, extending a data model is not done in a random way, but rather a certain logic and formalism must be followed and maintained. Extending a model in general creates heavy

files, and unless these files are handled through a database, it would be rather cumbersome to use them for advanced analysis. Versions of the model may change over time, and new classes may be added, updated or deleted. Changes come to support direct integration (level 2) with data related to specific applications or generic contexts (example of the “Dynamizer” concept of CityGML). Extended models need to be supported by a wide variety of tools. Unfortunately, this is still limited. Hence, integrating at the data model level is not simple, and solid knowledge is required to achieve the intended goal. Considering level 2, the main advantage is the straightforward data model compatibility. The data model supports, by default, the data to be integrated. For instance, the current implementations of UDT that are based on the CityGML data model have native support for almost all urban objects. Integration could also be performed using generic objects and attributes provided by the data model. Moreover, the new version of the data model enhances direct integration of the data into the model without further modification or extensions. However, this integration at the database level might require data preparation and data conversion. Indeed, the data may not be used as delivered, but some conversion workflows are mandatory to meet the model requirement. This leads to semantic data loss, and more specifically, during the mapping process between entities and attributes, which could be tricky.

We can state that levels 1 and 2 are both traditional levels of integration that have existed for decades under the auspice of many similar concepts. However, level 3 has been gaining high interest with the emergence of the UDT. Level 3 supports data integration through software and platforms. The integration at this level enhances syntactic interoperability. These architectures are based on separate and independent database management systems, which facilitate DT update. In addition, recent implementations show the possibility of running on-the-fly simulations based on web services and processes. The integration does not necessarily require a data conversion process since the current platform allows the handling and integration of various data types and formats. Level 3 also has some limitations, namely limited interactions, slow server response times, and poor rendering performance. The DT implementations that integrate the data at level 3 are generally tested in small areas (i.e., precincts and district levels). However, incorporating data using the same workflows at the city level requires optimizing the management system and visualization tools of such massive data. Recently, many studies have been carried out to effectively render and represent 3DCMs on the web. Hence, OGC standards and formats, such as 3D tiles and Indexed 3D Scene Layer (I3S), were developed to facilitate the streaming of these 3D models on the web. However, this format conversion from, for example, the CityGML 3DCM as a common input data to UDT to other web-friendly formats led to semantic data loss (Padsala et al., 2023). Furthermore, the conversion from CityGML to the web streaming format raises update issues, i.e., the integration of datasets coming, for example, from simulation software needs to be processed whenever new results are made available, and the updated CityGML model must be again converted to the required web data exchange format for visualization.

3.5 Unpacking and illustrating data integration levels according to the DT lifecycle

In the framework of this chapter, we intend to explain the data integration levels based on the DT lifecycle stages and analyze them accordingly (refer to **Figure 3.3**). The lifecycle

is defined by three primary stages: creation, which involves the generation of a UDT; usage, which focuses on the deployment of the fundamental component of the UDT namely the 3DCM in simulation applications and urban analytics workflows; and lastly, the update phase, which serves to continually maintain the UDT. We therefore conduct a more detailed analysis of these data integration levels across lifecycle stages, data types, and both current and potential scenarios. We have included the scenario dimension to highlight the various integration level options currently available and those that might be feasible in the future. This scenario exemplifies a UDT implementation which uses multiple data sets, and which can incorporate those data sets based on different levels of integration. By default, we have covered all the possible combinations of these approaches, leading to the identification of different scenarios. These scenarios will be further illustrated whenever possible by current initiatives and implementations.

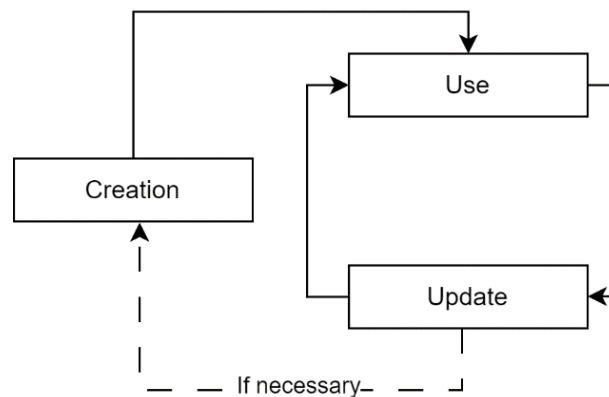


Figure 3.3. The Urban Digital Twin lifecycle stages.

3.5.1 Creation of UDT

The **creation phase** is defined as the first step towards the implementation of UDTs. This phase consists of the conceptualization of all relevant components, considering the core dataset to be integrated (usually geo-data), the modeling approaches, the required level of detail, and the representation and structuring of both data and models. Given that, in practice, the 3D city model is a part of UDT and not UDT per se, we will focus on this creation phase on the enrichment of 3DCMs as an integration foundation for UDT. In practice, creating UDTs is often seen as building a semantic city model or connecting different data models and systems into one unified model. The creation phase in this study implies the enrichment of existing city models or systems to meet the DT's requirements since most of the implementations are not built from scratch. This is achieved concretely through data integration.

To create a UDT, data integration can be performed at different levels, as defined earlier in Section 3.3. Our focus is not on data-to-data integration (also known as data fusion), but we are particularly interested in data-to-model and model-to-model integration. In addition, the seamless integration of various urban city objects to create a complete and comprehensive 3DCM needs to be considered as well.

To clearly understand the three levels of integration, we extract the common scenarios for creating a UDT based on the literature. **Figure 3.4** gives the various current scenarios of data integration levels in the creation phase.

Phase 1: Creation - Current scenarios			
SC \ Lv	Lv1	Lv2	Lv3
A			
B			
C			

Figure 3.4. Current scenarios of data integration levels in the creation phase.

Scenario A: This involves integration at level 1. This requires changing the model if it does not handle the integrated data by default. In this regard, CityGML is widely adopted as a data model for semantic 3DCMs, and recently, UDTs. Thanks to its common urban object native support, its extensibility and its interoperability, it is considered a basis for data integration to create a comprehensive and accurate UDT. This creation could be achieved by augmenting its data model with new and additional concepts using generic ADEs. For instance, generic ADEs are implemented to adapt the data model to support a national standard or to cope with specific needs.

Most CityGML models are based on version 2.0, which probably uses ADE to handle and integrate new datasets and thus tackle new problems. With the new version 3.0, this integration has changed, for example, the “Dynamizer concept” has a direct compatibility with the core model, thus enabling data integration at level 2 according to our definition. In this regard, new versions of data models may introduce new classes and attributes to the schema or remove some parts of them to fit the UDT requirements. This ultimately affects the level of integration.

Scenario B: Integration at the database level involves the use of an ETL process to feed data into the model. The database level is commonly used in UDT implementations based on the native data model or through the support of generic objects and attributes.

Scenario C: Data integration at level 3 is relevant in the creation phase. The integration takes place at the front-end level. The model remains untouched, and the databases are completely independent. The data is integrated using web architecture or game engines. Table 3.2 further illustrates the different levels of data integration. It provides current illustrative DT implementations that are in the creation phase and use different data integration levels.

Table 3.2. Examples of data integration levels (Lv) in the creation phase using different data inputs.

Data input	Creation phase		
	Lv1 (Scenario A)	Lv2 (Scenario B)	Lv3 (Scenario C)
Static geospatial data	- CityGML 2.0 ADE of the Vienna City (Lehner et al., 2024)	- Mapping the data and their attributes into the 3DCityDB tables (Diakite et al., 2022)	- Cesium Ion (Lei et al., 2024) - Cesium Ion, VR and JS libraries (La Guardia et

			al., 2022; La Guardia and Koeva, 2023)
			- COVISE (Collaborative Visualization and Simulation Environment) (Dembski et al., 2020)
Other static data	- IFC-CityGML data model (Li et al., 2020, 2019)	- CityGML 3.0 (space concept) (Kutzner et al., 2020)	- Integration of both BIM data and geodata in the same front-end (i.e., Esri ArcGIS CityEngine, ArcGIS GeoBIM, virtualcitySYSTEMS)
Dynamic data	- Dynamizer ADE (CityGML 2.0) (Chaturvedi, 2021; Chaturvedi et al., 2017) - Extending 3DCityDB by Dynamizer ADE (CityGML 2.0) (K. Chaturvedi et al., 2019) - Dynamizer (City)JSON 1.1) (Boumhidi et al., 2024)	- Generic attributes (Chaturvedi, 2021) - Dynamizer module (CityGML 3.0) (Kutzner et al., 2020)	- CityThings (connection between the SensorThings API (STA) and the 3DCM using gml-id) (Santhanavanich and Coors, 2021) -COVISE (Collaborative Visualization and Simulation Environment) (Dembski et al., 2020)

Many examples might be used to illustrate these levels during the integration phase. To illustrate our statements, we discuss below several related research under the prism of our classification.

Based on Table 3.2, Level 1 is used in the preliminary phase of creating the DT of the City of Vienna (Lehner et al., 2024). To support the long-term vision of creating a DT of the city, an extension of the CityGML 2.0 data model profile is being performed to meet the current and future requirements of the municipality. The data model is augmented with additional features using ADE to create a complete 3D model of the city. For level 2, a study case was performed within an Australian DT pilot project (Diakite et al., 2022). The authors integrate the spatial data sets provided using a database approach. They process, incorporate, and store existing 3D data in the 3DCityDB using PostgreSQL and PostGIS database, the main core of their DT architecture. The data went through a mapping process to meet the CityGML classes and attributes since the data were provided in different structures. Hence, the direct import using the official 3DCityDB Importer tool of the data sets to the database was tricky. In this regard, a series of Python algorithms were developed to reach the integration goals. To illustrate level 3, recent related work designs a socio-technical conceptual framework that illustrates the need to incorporate human

sensing data (i.e., participatory data) into UDTs (Lei et al., 2024). In their case study, the authors implemented their UDT prototype based on multiple open data sources (static and dynamic data), namely, crowdsourced data (e.g., Open Street Map data), participatory data (social sensing data), and environmental data. Data integration was performed using Cesium. The platform is based on browser-server architecture, fostering data interoperability and system compatibility with current and upcoming input data and systems. The data went through a data conversion process based on an FME Flow. The platform supports several formats, such as 3D tiles and JSON-based formats, as well as Cesium Markup Language (CZML), that represent human sensing data interlinked with the geospatial component. Another related work proposes a 3D data fusion workflow based on Virtual Reality and JavaScript library-based 3D WebGIS solutions (La Guardia et al., 2022; La Guardia and Koeva, 2023). In their work, the authors handle and integrate different and complex geospatial input data, namely point cloud data, 3D models, and BIM models.

The same logic applies to BIM-GIS integration, which is considered one of the fundamental data integration challenges while moving towards implementing such a transdisciplinary concept as UDT. This integration could be performed at level 1. For instance, a study focuses on the precinct scale and develops a semantic precinct information model (PIM) using multisource and various data (Li et al., 2020, 2019). Based on the conceptual and unified data model, the authors implemented a relational spatial database to manage the building model. The integration at level 2 is feasible by mapping IFC classes to CityGML classes and attributes, as well as with the last CityGML 3.0 (Kutzner et al., 2020). Data integration is achieved at level 3 via GIS and urban tools (i.e., ESRI CityEngine, ArcGIS GeoBIM, virtualcitySYSTEMS).

Integrating dynamic data (such as sensor data) highlights the key features of UDT. Indeed, this data integration within UDT can be achieved by extending the data model concept, as seen in "Dynamizer ADE" for version 2.0 (Chaturvedi et al., 2017). Alternatively, integration can occur at level 2, exemplified by the use of the thematic module "Dynamizer" in CityGML 3.0 (Kutzner et al., 2020) or sometimes using generic attributes (Chaturvedi, 2021). At level 3, integration happens through system architectures, notably by ensuring a connection between the server handling the sensor data and the 3DCMs (Santhanavanich and Coors, 2021).

Herrenberg's city-scale DT prototype is a project that illustrates the heterogeneous data integration (i.e., static and dynamic data) in the creation phase, where various existing volunteered geographic information and a 3D city model are used. Furthermore, thematic data such as mobility data from mobile applications and GPS (i.e., movement traces of bicycles and pedestrians) are used. The integration is performed at the front-end level. The project uses COVISE (Collaborative Visualization and Simulation Environment), an extendable distributed software that supports multisource data and sensor network data as well, resulting from a computational model for air pollution simulation.

In this regard, data integration in the creation phase could be performed according to different levels of data integration. As discussed earlier, there are current scenarios that are implemented in practice. However, expected theoretical scenarios in the context of creating UDT need further investigation (see Figure 3.5). Indeed, it is possible to combine different levels of integration. This leads to the following integration scenarios.

Phase 1: Creation - Theoretical scenarios			
SC \ Lv	Lv1	Lv2	Lv3
D			
E			
F			
G			

Figure 3.5. Theoretical scenarios of data integration levels in the creation phase.

Scenario D: Integration in this case might explore levels of integration 1 and 2. For instance, road data is provided and needs to be incorporated into the UDT. Level 2 might be used to convert the data to meet the requirements of the data model, which could also be extended and modified (level 1) to receive the transformed data.

Scenario E: Levels 2 and 3 are used in this context. For instance, the data model will natively handle the data to be integrated, but additional data is not supported. Thus, integrate them using web architecture or game engine workflows.

Scenario F: It involves the use of levels 1 and 3. Data could be integrated based on level 1, and further data in the same UDT implementation uses the front-end level to integrate the data.

Scenario G: This scenario is the most faithful to the definition of a generic DT, in which a DT has a concrete data model, a database that manages the corresponding objects, and, finally, one or several client applications. With this in mind, the DT is neither a file- nor database-based system but can be assimilated into a three-tier architecture (model/DB-server-client).

Implementations D, E, F, and G are proposed in this work as a combination of different levels to integrate data into UDT. Yet, such implementations are hard to find and should be investigated in the future.

From practice, the creation phase is mainly focused on the data enrichment of the available 3D models to generate an accurate and complete digital representation. Enrichment is performed based on different levels of integration. Based on our investigations in the creation phase, we can clearly conclude that level 2 based on the CityGML 2.0 data model is the common data integration level. Level 1 is explored to meet the national DT requirements using ADE in this preliminary phase. However, this needs further investigation in the future. Level 3 is explored as well by connecting different input data at the front-end level.

3.5.2 Use of UDT

The **use of the UDT** corresponds to its operation in a specific use case to address a particular application. In practice, this stage can sometimes be confused with the creation phase, particularly when the creation of a DT is based on a use case, also called a UDT-based use case. Nevertheless, the use of UDT is defined as the process of performing simulations and urban analysis. This phase is closely related to the use case requirements,

where input data is required to generate new data sets or models, which in turn will be re-injected into UDT to support decision-making. The produced data sets in this phase might be incorporated into the DT following various data integration levels. While these integration levels have been experienced with 3DCMs for multiple use cases and applications, this section will tackle this integration aspect within the context of UDT available implementations.

Within the geospatial community, 3DCMs form the foundation for these data integration levels in the UDT scope. In this regard, and considering current and ongoing UDT implementations, we intend to outline and discuss the levels of integration identified in the literature.

Scenario A: This scenario is also known as the traditional way of integrating data. It involves the extension of the CDM to manage and store the data generated from the simulation tools into the 3DCMs. Indeed, ADEs, namely Energy ADE and others, are used to perform this integration task (Agugiaro et al., 2018). Moreover, with the growing interest in breaking down data silos, defining a standardized open data model is thus of great interest. For instance, studies investigate the possibility of designing a Food-Water Energy ADE (FWE ADE) (Padsala et al., 2021). The approach is based on extending the CityGML data model to support the integration of food, water, and energy data and simulation workflows. The approach is tested at different spatial levels with different input data. The results are stored in the added module as defined in the FWE CDM. While moving towards sustainable cities, attempts to manage and store multi-domain data are still limited to a few implementations.

Scenario B: Level 2 is explored in this scenario. The model handles by default the simulation outputs, measured data, or any other data produced by simulations and urban analysis or provided by third parties.

Scenario C: This level requires system architecture that allows bottom-up data integration of urban information and 3DCMs (associating separate database management systems) as well as their visualization.

Scenario D: This approach involves data integration based on levels 1 and 2. The combination of levels 1 and 2 consists of adapting the data model to integrate the data and, at the same time, using level 2 to enrich the model where applicable. For example, extending the CityGML model to incorporate simulation results and using generic tables to integrate further data.

Scenario E: Levels 2 and 3 are used to integrate data into the UDT. This scenario takes advantage of level 2 since the data model might by default support data integration for a given use case; at the same time, it uses level 3 to enrich the UDT with various urban data.

Scenario F: This scenario involves the use of both levels 1 and 3. This implies that a data model is adapted for a specific use case and an additional system architecture is designed to support the integration.

Scenario G: This scenario implies the use of the three levels of data integration. Each level is deployed to integrate a specific data type on the same UDT platform.

To illustrate current scenarios, we adopt the energy use case. The use case was selected on the basis that many research studies have already identified and tested the different levels of integration. Thus, we extracted the current scenarios (refer to Figure 3.6) and analyzed the references linked to this UDT based on data types (refer to Table 3.3) and identified theoretical scenarios (see Figure 3.7).

Phase 2: Use - Current scenarios			
SC \ Lv	Lv1	Lv2	Lv3
A			
B			
C			
E			
G			

Figure 3.6. Current scenarios of data integration levels in the use phase based on an energy use case.

Table 3.3 identifies the different levels of data integration. It provides illustrative DT implementations that are in the use phase. The implementations use different data integration levels.

Table 3.3. Examples of data integration levels in the use phase based on an energy use case.

Data input	Use phase		
	Lv1 (Scenario A)	Lv2 (Scenario B)	Lv3 (Scenario C)
Energy simulations	- Energy ADE (Würstle et al., 2020)	- Extended 3DCityDB with Energy ADE tables (Rossknecht and Airaksinen, 2020)	
Measured data	- “Dynamizer” ADE (Chaturvedi, 2021)	- Extended 3D city DB with generic tables (Würstle et al., 2020)	- CityThings concept (Santhanavanich and Coors, 2021)
Lv2 and Lv3 (Scenario E)			
Static properties and time series datasets simulation	-Static properties/Attributes (generic attributes) -CityThings concept (T. Santhanavanich et al., 2022)		
Lv1, Lv2 and Lv3 (Scenario G)			
2D geometry and attribute data/Temporal sensor data/ Urban building energy data	- Energy ADE schema (Building Energy and Climate Atlas)- Enrichment of CityGML data model with attributes of building function and year of construction- OGC API – Features- STA service - OGC API Processes services (Santhanavanich et al., 2023)		

Based on Table 3.3, current scenarios are identified as follows:

Scenario A: Based on CityGML 2.0, an energy-related urban data model is defined to calculate and enrich the data model with building energy information required for energy simulations (Würstle et al., 2020). Hence, Energy ADE has helped in addressing data interoperability issues. In addition, "Dynamizer" ADE is also used to manage measured data in the energy use case (Chaturvedi, 2021).

Scenario B: Extending the 3DCityDB to support Energy ADE allows us to perform data integration at level 2 (Rosknecht and Airaksinen, 2020). For instance, energy information is stored in 3DCityDB, where Energy ADE tables are already embedded. The aim of this database approach is to directly feed and update the database tables from the simulation results without the need to alter the CDM. The results are written back to the database, escaping the challenges related to the file-based approach. Alternatively, integration at Level 2 might be carried out through generic tables (Würstle et al., 2020).

Scenario C: For instance, the "CityThings" concept enables managing and integrating dynamic sensor data and 3DCMs provided by different parties in a separate management system (Santhanavanich and Coors, 2021). The concept connects sensor data from SensorThings API server and the 3DCM. The approach addresses the challenges related to the direct storage of high-frequency measured data into the CityGML data model (either for file- or DB-based approaches). In fact, level 3 is designed when the data model does not allow integration or represents some limitations; hence, the adoption of system architectures and the APIs standards at the front-end level facilitates integration and maintenance within the DTs framework.

Scenario E: An illustrative use case of this level of integration is provided in the work done by Santhanavanich et al., (2022). The authors propose a method to tackle the traditional data integration workflow of the energy simulation results into the 3DCM before converting the new CityGML model to a ready-to-web stream format. The aim is to optimize the data management of 3DCMs using SDI through web applications. In their SDI approach, the authors use a range of OGC web services standards, namely the SensorThings API, the OGC API 3D GeoVolumes, the OGC API Features, and the CityGML standard. The simulation outputs are handled differently; the static building properties are integrated directly into the CityGML data model (which refers to level 2). However, the spatiotemporal data is stored and managed in an external database. This was performed at level 3 based on the STA exchange format specification by ensuring a connection link to the OGC web services.

Scenario G: This scenario implies the use of the three levels of data integration. Each level is deployed to integrate a specific data type on the same UDT platform. This scenario was recently implemented in a study by Santhanavanich et al. (2023). The authors illustrate a web-based API workflow to integrate and handle geospatial data, 3DCMs, building energy data, and simulation results based on a web service and OGC API standards. The aim of the approach is to create a single and harmonized data model combining geospatial and energy data that are in practice provided and maintained by different providers. The approach involves various data integration levels. Energy ADE is explored, which illustrates level 1. Furthermore, the enrichment of the CityGML model is performed at level 2 through mapping approaches. Finally, the simulation results are integrated based on modular web architecture based on OGC API standards, namely OGC processes and the

STA. The overall web-based API architecture is tested during OGC Testbed 18²⁴ for two study use cases (i.e., Montreal and Helsinki). The aim of the testbed is to extend the geospatial spatial data infrastructure to support building energy data interoperability as well as setting the fundamentals of energy spatial data infrastructure. The SDI addresses interoperability issues and integration challenges through standardized web services and interfaces. Data visualization is also performed successfully using innovative tools to facilitate data access and analysis (WebGL and AR applications). The testbed-18 pilot project aims to generalize the workflow to the entire city using a high-level developed prototype. Furthermore, the participants highlighted the potential of moving towards a generalized data model. This data model seeks to structure and consolidate both geospatial and energy datasets based on a standardized modular web services-based architecture.

Scenarios D and F are identified as theoretical scenarios. They have been less explored in the framework of integrating data into UDTs (see Figure 3.7).

Phase 2: Use - Theoretical scenarios			
SC \ Lv	Lv1	Lv2	Lv3
D			
F			

Figure 3.7. Theoretical scenarios of data integration levels in the use phase.

Scenario D: Given that we are interested in the UDT-based energy use case, we were unable to find an illustrative study case that allows us to further explain the integration process.

Scenario F: In this case, a suitable integration-based energy use case was hard to find.

Based on the previous analysis, the use phase is closely driven by the use-case requirements. This phase encourages collaboration and convergence between various domains. However, the heterogeneity and complexity of the data in this phase raise data integration challenges. In this regard, data integration levels have shifted increasingly from file-based approaches to standard web-based API approaches. The new trend towards data integration at level 3 has been involved in recent UDT implementations. It is considered the most suitable level to integrate any type of data. However, the data, namely geospatial data, needs to be FAIR (findable, accessible, interoperable, and reusable), and domain-specific data should be provided in a standard-compliant format. In addition, the data needs to be well documented to facilitate the integration and maintenance processes with formatted metadata and sufficient technical documentation. This mapping of integration levels, as detailed above, can be extended to cover any use case. This analysis of the levels based on the use case enables us to identify the patterns of data or model integration according to the urban application. It is also worth noting that the use phase can serve as a guide for the creation phase. For example, in the creation phase, a semantic 3DCM was built based on level 1. In the use phase, it appeared that

²⁴ https://docs.ogc.org/per/22-041.html#_0998bc4e-baa2-49b0-901c-ae2572a41b8f

model enrichment was required to run the simulations. This will probably prompt level 1 or 2 of data integration to generate the necessary data for the use phase.

3.5.3 Update of UDT

The **update phase** is an extremely valuable step in the lifecycle of UDTs. This phase is unfortunately overlooked in current implementations. Nevertheless, it is always highlighted as one of the key challenges to be considered when maintaining DTs. The UDT should be conceived as a sustainable process and not as the final product. For this reason, the definition of integration approaches is an essential requirement in the updating phase. That is how these different levels have their merits and demerits. Proper strategies and mechanisms to support the update of UDTs are still limited. This issue has already been experienced with 3DCMs. Although various DT parties (i.e., geodata providers, researchers, and municipalities) are aware of the necessity of data maintenance and governance. Current implementation still lacks concrete investigations. Given that performing updates is tightly related to data integration, we will discuss in this section the data integration levels in this phase of the lifecycle.

Updating a UDT according to the three basic levels as previously defined can be envisaged (see Figure 3.8); however, a scenario-based analysis, as conducted in the creation and use phases, is not currently going to be carried out, given that the update mechanisms are limited to a few DT prototypes.

Phase 3: Update - Basic scenarios			
SC \ Lv	Lv1	Lv2	Lv3
A			
B			
C			

Figure 3.8. Scenarios of data integration levels in the update phase.

One of the main characteristics of UDTs is that the virtual system is updated regularly through the bi-directional link between the real and digital worlds. Hence, data integration methods should be set to guide the user while integrating new data or information. The new data could be obtained from different sources and cover different themes. However, the appropriate level of data integration is guided by the previous phases (i.e., creation and use).

Data maintenance logic should consider the data, which is the fuel of the DT, and the DT components, which structure these data. In this respect, it is vital to distinguish between the updating of data, which may be handled by the data provider, and the updating of DT's components given a new set of data, which does not solely rely on the data but rather encompasses the techniques and methods applied in the development of these digital twins.

To illustrate, let us consider the updating of a 3D city model, which is one of the main components of Digital Twins. Possible updates in the creation phase are, for example, a

new LiDAR acquisition, a new building created, a road removed, or new vegetation, to name a few. These data are generally updated by their providers (notably the cadastral department). Although this is done systematically, however, how do we update the UDT? There are several possibilities for a new LiDAR acquisition: (1) A new LiDAR acquisition implies a new model and, therefore, a new integration of the model in the UDT (re-modeling). (2) compare the new point cloud acquisition with the old one using change detection methods, then regenerate the 3D model. (3) compare existing models with the point cloud, then update the model. (4) compare based on model-to-model approaches and explore versioning of 3D city models. In all cases, these possibilities will require the reintegration of the 3D city model into the UDT. Furthermore, the update mechanism is related to the data integration levels used in the creation phase. Indeed, if the creation phase is based on the conceptual data model level, the update would be performed at level 2, given that the core model used in the creation phase handles these urban objects (see Figure 3.9.a).

The same logic is applied to the usage phase. For instance, to update specific data that was integrated into the use phase (i.e., when new simulation workflows are run based on new datasets or parameters), the new outputs should be injected into the UDT, and the attributes need to be updated accordingly (refer to Figure 3.9.b). Generally, level 3 is the most convenient data integration approach for dynamic databases, such as the one explored in the UDT in the use phase. This level seems appropriate for dynamic data such as the one measured from sensor data as well as the simulated outputs. The particularity of these data is that they are generally managed in separate systems in the current implementation. Thus, updating them is performed in a straightforward manner. Furthermore, Level 3 appears to be a promising approach for data updates, since data are maintained by their respective owners and shared as services through web-based architecture, enabling distributed and up-to-date access. This increases access, querying, sharing, and updating of the data. Furthermore, the update could be performed on the front-end side by means of user interfaces.

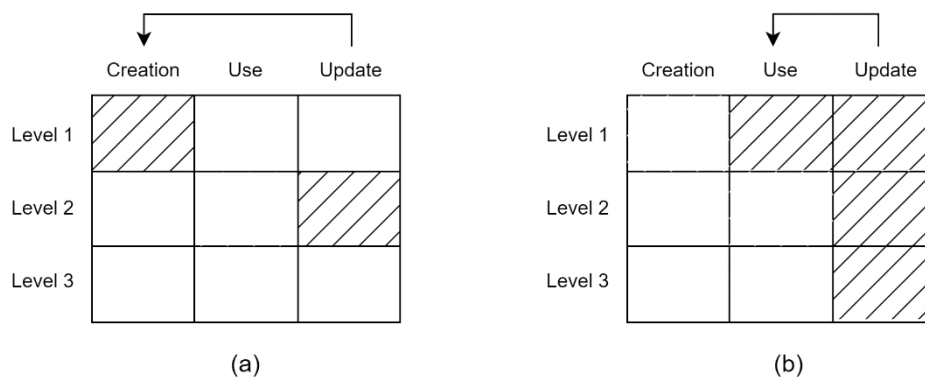


Figure 3.9. Data integration levels for DT updates: (a) at the creation phase, (b) at the use phase.

Although we have already outlined the basic update scenarios, in this phase, the main question to answer is which part of UDTs needs to be updated, so the data integration levels are decided accordingly. The data deployed in the update phase needs to conform to the UDT requirements. Furthermore, standardization is mandatory to detect changes

in the data and facilitate their maintenance. It is also important to highlight that updates are usually more focused on the use phase than on the creation stage. This is because the whole purpose of a UDT is to create it only once, to use it in multiple use cases, and to perform updates when appropriate, especially as this is a costly process in terms of human and technical resources and will depend on the availability of data.

3.6 Data integration levels: discussion and findings

Data integration is considered both a fundamental requirement of UDT and a key challenge that hinders its full implementation in practice. Different models, architectures and platforms are developed and named UDTs. However, they represent discrepancies in terms of how the data is integrated. To properly understand the data integration levels, we represent these levels according to the DT lifecycle inspired by the geospatial data lifecycle. Nevertheless, geospatial data are not the only kind of data to be incorporated into DTs; various domain-specific data can also be integrated (i.e., energy data, noise and air quality data, computational fluid dynamics outputs).

This integration aspect is involved in all DT stages, from creating DTs to the use phase (simulations and what-if scenarios), and finally to maintaining the twin.

Current UDTs are mainly based on semantic city models. These models are considered the foundation for data integration. However, if we change this model and extend it, we are performing integration at level 1. On the other hand, when the default model supports the data to be integrated, we are at level 2. However, if we add architecture components to embed the data, we are at level 3. The action of integrating enables model enrichment to create comprehensive DTs of the real world. The creation of DTs needs to be discussed upstream to ensure that models can be maintained. In fact, current implementations use existing models and adapt them in certain ways to fulfill DT requirements. In the DT creation phase, we refer to the creation of city models while emphasizing that these models are developed to form the basis of the UDT. Indeed, we are lacking a generic DT that can be relevant when creating regional or national DTs. At the city level, however, we tend to focus on implementations based on use cases, which lead to the generation of multiple DTs and, by default, affects the integration levels. We have noticed that in the creation phase, we usually use semantic models of current cities, such as Helsinki, Vienna, and Japan, to name a few. These models generally integrate data based on levels 1 or 2 in the creation phase. Even in the use phase, ADEs, or generics (objects or attributes), are deployed. Other implementations make extensive use of level 3 to integrate data, but more precisely during the use phase.

In the present work, we wanted to highlight the data integration levels and offer some conceptual guidelines according to DT's lifecycle to enhance data integration and interoperability. Data integration enables DT practitioners to shift from a semantic city model to a UDT. Integration takes place at different levels, and various configurations of levels can be envisaged, notably in UDT, where several types of data are required. For each phase of the DT lifecycle, we integrate data, and for each phase, one or more levels of integration can be applied.

Moving from a city model to a UDT, DT designers need to plan the integration of different types of data. The best level of integration does not exist. In fact, each UDT project has its own appropriate integration level, and this is also guided by the DT lifecycle phases. For instance, in the creation phase, it is important to consider that these models will be used in simulations and urban applications, which in turn generate new data that needs to be reincorporated. In this regard and following the analysis provided in Section 3.5.1, level 3 in the creation phase is still limited to few implementations given that UDT developers tend to use level 1 of integration based on the conceptual data model. This is a valid assumption since the core of Urban Digital Twins is a 3D city model that represents the foundation layer of integration from geospatial perspective. However, in the use phase, level 3 is commonly used. To support integration at this level, database-independent architectures are developed.

For the update phase, we are unable to draw any conclusions on the appropriate level of integration, given that in the current implementations, we were unable to identify any proposed mechanisms for updating UDTs. This is also true since almost all current implementations are still in the creation and use phases. However, to overcome this, we would like to set up experiments for future work to understand how the DT parties conceive this UDT update.

While this study helps to provide fundamental insights regarding UDT data integration challenges, however, it has some limitations that will be tackled in future work. One of the limitations is the selection criteria we have used to filter projects. To illustrate our data integration levels, we examined DT projects in detail, which was not a straightforward process since DT practitioners do not always describe the way in which they integrate their data and whether they are really considering a generic and standardized integration approach to homogenize their data. Another aspect is that the current initiatives are mainly focused on the creation and use stages of digital twins, yet there is a minimal distinction between the creation and use phases. Furthermore, our framework is conceptual and uses real-world implementation to prove its findings, which raises the issue related to the feasibility of implementing all the integration scenarios previously defined in future work.

In this study, we have focused on the most common datasets, such as geodata and simulation outputs. However, taking into consideration the advancements in Generative AI, such models will in their turn bring new integration challenges. At the current maturity stage of UDT, these models help to address issues related to the data quality, availability, and completeness by generating comprehensive and realistic data through data enhancement and enrichment. They are generally used in urban studies to produce synthetic data, automate the processing in terms of semantic segmentation, and generate 3D city models.

3.7 Conclusions

Cities worldwide start investing in developing UDTs as a new digital approach to collecting, integrating, managing, visualizing, and sharing data and models. UDTs bring both potentialities and issues. Data integration is identified as a significant challenge associated with the full implementation of the UDT concept. While data integration has been considered a fundamental issue in 3D GIS for a decade, it is still involved in the development of new concepts such as UDTs, which integrate multiple and heterogeneous data. Indeed, addressing the data integration issue will enrich the existing 3D city models and deliver a mature version of the DTs. In this chapter, we provided the first conceptual data integration insights to create, use, and update a UDT. Three data integration levels (conceptual data model, database, and front-end levels) are defined and illustrated according to the DT lifecycle. For each phase of the lifecycle, the data might be integrated using one or several levels on the same UDT platform. The conceptualization of the data integration levels is a domain- and application-independent classification. That means that the classification proposed in this work might be applied to any UDT implementation. Furthermore, we believe that new data integration levels will appear in the future to handle UDT data and models. Regardless of the advantages and limitations reported for each data integration level, data quality, completeness, and interoperability strongly guide the effectiveness of the integration. Our defined data integration levels will help researchers and practitioners (namely cities and municipalities) that have already an ongoing UDT implementation or those who are willing to embark on this journey towards the creation of urban digital twins to understand and transfer the common practices related to data integration and to discuss upstream the integration frameworks, the level(s) required according to the available data and the user needs. The users can be government, researchers, private sector, and finally citizens.

In future research, we will focus on technically testing the different scenarios, investigating them based on the DT lifecycle, and formulating guidelines to recommend effective data integration levels based on different standards and frameworks. Future steps aim to investigate in more detail each scenario defined in this work and test it in a specific case study using various datasets. We believe that implementing technically the different data integration configurations will help us to determine their relevance, evaluate the robustness and applicability of the generic data integration levels in different use cases, and assess the most effective level of integration. To reach our intended goal, we should establish a technological framework and provide solutions in terms of data models, databases, and visualization tools, which will accordingly assist the development of the UDT. An exploratory survey will be designed in the future to get insights from UDT practitioners about their thoughts regarding these data integration levels and if their initiatives follow any of these integration levels. The focus will also be on evaluating the differences of data integration in the maintenance phase. This will bridge the current gap in literature in terms of how integration is carried out in the maintenance stage.

3.8 Subsequent work

Building upon our previous work, which proposed a comprehensive conceptual framework defining levels of data integration, this work seeks to gather insights from practitioners actively engaged in UDT implementations. The aim is to understand how these conceptual integration levels are applied in practice, how they are selected according to the lifecycle, what tools support their implementation, and how they enable effective integration depending on the availability and nature of the data. By aligning theoretical frameworks with practical implementations, this research aims to enhance the effectiveness and adoption of scalable and flexible data integration strategies in the development, use and maintenance of UDTs. The findings highlight the relevance of data integration levels for UDT practitioners, demonstrating that they were able to identify which levels are involved in their projects. Additionally, the results provide insights into how each level of integration is commonly applied across different stages of the lifecycle, as well as the tools and methods associated with each level.

The participant's profile is presented in **Figure 3.10**. Most of participants are from Academia (24), 7 from Government/Polymaking and 4 from Industry. Participants' field of expertise is namely from Geoinformation with 30 experts. The participant's geographical location is deduced from the project initiatives with most UDT projects conducted in Europe (22). Many European countries have reported their initiatives, including Netherlands (6), Germany (6), Switzerland (1), Finland (1), Poland (1), Austria (1), Sweden (1), Croatia (1), Slovenia (1), Bulgaria (1), and the United Kingdom (1). Additionally, two projects were reported at a European scale by participants from Italy. Beyond Europe, we had 7 participants from the Asia-Pacific region, including Singapore (2), Australia (3), and South Korea (2), as well as 4 participants from North America, comprising (3) from the United States and (1) from Canada.

Furthermore, participants provided additional information about their UDT projects, namely regarding spatial scale and lifecycle stage. To capture project scale, we asked respondents to select one predefined option (National, Regional, City, District, Quarter/Block), which enabled us to categorize initiatives and better understand how UDT implementations are distributed across spatial levels. The results show that most initiatives operate at a city-wide scale (13), followed by national (9) and district (6) levels. In addition, seven initiatives reported other scales, including Global (1), European (3), and Campus (1), as well as multiscale approaches such as projects spanning multiple municipalities (1), testbeds (1), and city blocks (1). We also examined the UDT lifecycle, recognizing that UDTs are not static snapshots but dynamic systems that evolve through continuous integration and real-time data enrichment; identifying where initiatives sit within the lifecycle helps assess maturity and operational status. Our analysis indicates that 18 initiatives are in the creation phase (currently under implementation), 13 are in the use phase (operational), and 4 are in the update phase. This distribution underscores the progressive development of UDT projects, with a significant number transitioning from conceptualization to active use and continuous maintenance. Participants were asked to specify the integration level(s) used in their projects. (16) projects use level 1, (20) choose level 2, and (16) adopt level 3. It is important to note that Figure 3.10 represent individual counts per level; some projects may utilize multiple integration

levels simultaneously. Therefore, the total number of projects across all levels may exceed the actual number of the 35 initiatives. Additionally, some participants were unable to clearly identify or select a specific integration level that aligned with their implementation. However, by selecting the "Others" option, they were able to provide additional explanations on how they approach data integration. These responses reflect alternative approaches and methods. For example, some participants expressed uncertainty about the question and a lack of clarity regarding the integration levels. One participant mentioned that their focus is not on data integration but on linking databases (geodata and management data) at the object level. Other participants indicated that different parts of concrete usage/adoption belong to different levels.

These findings underscore the diverse approaches to data integration among the surveyed initiatives and highlight areas where further clarification in survey design could enhance response accuracy. Furthermore, these responses from domain experts will help update the previous understanding of data integration levels in UDTs.

In this work, we also carried out cross-analyses of the participants' responses. These analyses investigate the relationships between project scale and lifecycle stage, as well as between lifecycle stage and levels of integration.

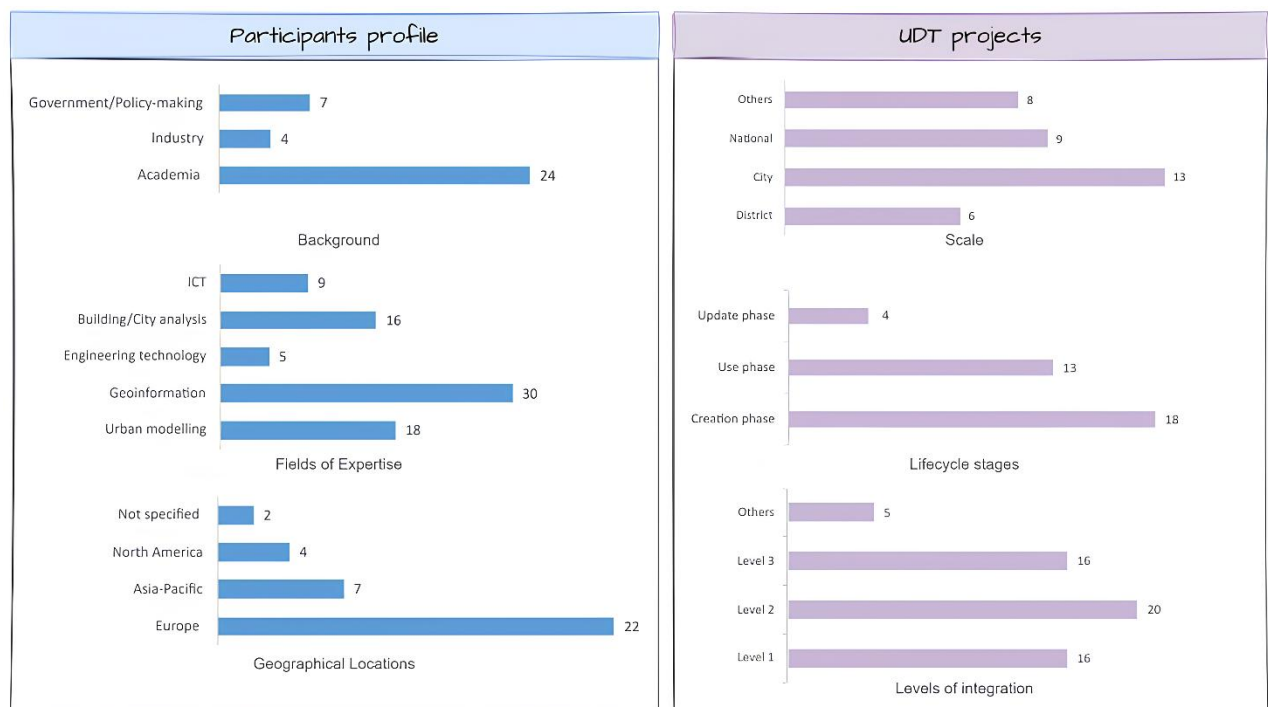


Figure 3.10. Overview of the participants' profile and the UDT projects.

We have presented the lifecycle stages per spatial scale and concluded that several initiatives are city-wide and span the entire lifecycle of the digital twin (refer to **Figure 3.11**). National-scale DTs are at the creation stage, although a few initiatives are at the usage or update stage. While for other urban scales, notably districts, blocks, and municipalities, they are more at the creation and use phases. From this analysis, we can conclude that efforts are highly concentrated at the city scale, with 15 initiatives, including 5 in the creation phase, 5 in the use phase, and 3 in the update phase.

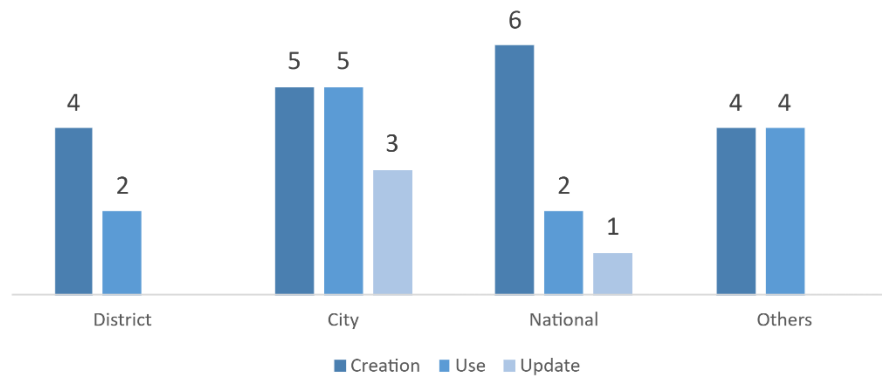


Figure 3.11. Comparative analysis of UDT projects: spatial scale vs. lifecycle stage.

To gain a better understanding of these integration levels and their distribution throughout the digital twin's lifecycle, we conducted a cross-analysis to identify how many projects use each integration level during each phase (refer to **Figure 3.12**). We conclude that projects in the creation phase predominantly rely on Level 1 and Level 2. In contrast, projects in the usage phase extensively employ Levels 2 and 3. Notably, in the update phase, Level 1 integration is absent. This suggests that if Level 1 was utilized during creation, subsequent updates are likely performed using Level 2, as the conceptual model can now accommodate new data, or Level 3, when external APIs are employed. Identifying these patterns highlights the importance of selecting appropriate integration levels at each phase of the lifecycle to ensure seamless data flow and system interoperability.

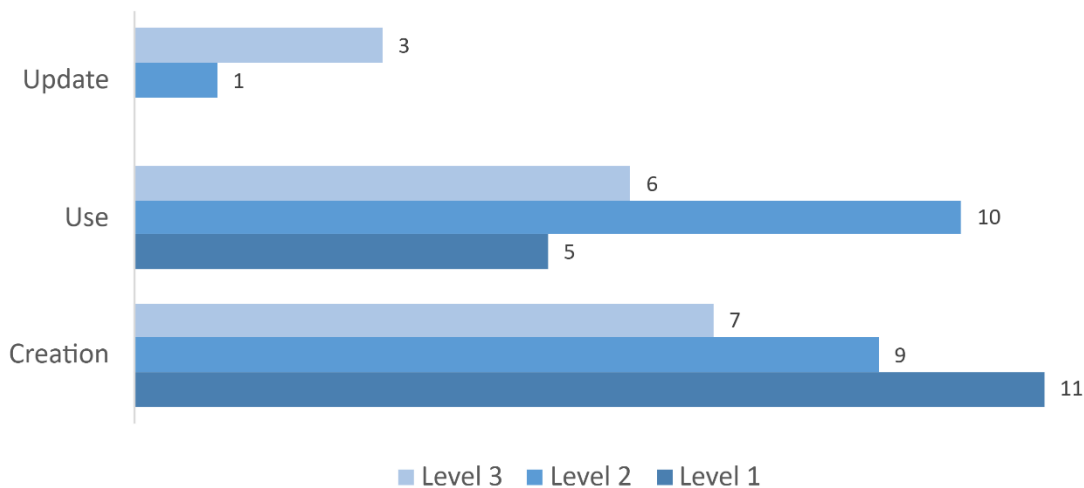


Figure 3.12. Comparative analysis of UDT projects: lifecycle stage vs. levels of data integration

We asked participants about the relevance of the data integration levels and whether they resonate with their own implementations (see **Figure 3.13**). 89 % of respondents from diverse backgrounds confirmed the relevance of the data integration levels. The high proportion of "Yes" and "Maybe" responses further emphasizes that, while some participants fully recognized the relevance of the levels (54% responses) (refer to Figure 3.11), others may have experienced partial alignment while reflecting their alignment with ongoing Urban Digital Twin (UDT) development efforts. Indeed, 34% of practitioners

responded "Maybe," indicating some level of uncertainty. However, these participants were still able to recognize the levels of integration deployed in their implementations, suggesting that the classification is both applicable and understandable, implying continued interest in the UDT development process. A minority (12 %) responded "No," highlighting potential gaps in how the classification aligns with the tools, workflows, or data practices used in their specific contexts.

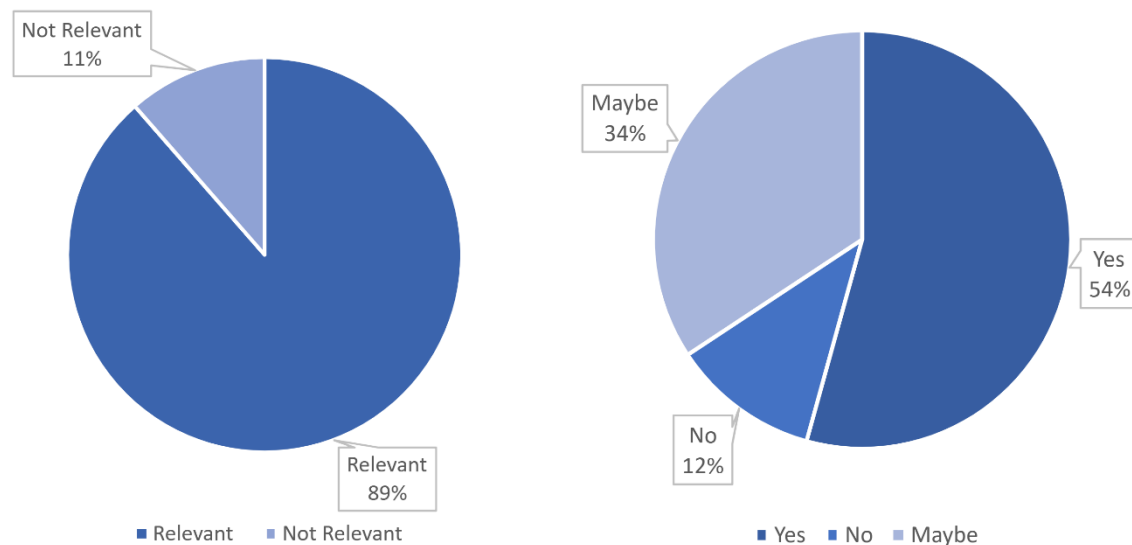


Figure 3.13. Relevance of data integration levels (a) among practitioners and (b) for UDT implementation

In the framework design of the survey, practitioners were asked whether they have set any proper strategies and mechanisms to support the update of their UDT. While the challenge of maintaining UDTs should ideally be addressed earlier, even before any actual updates occur (e.g., while designing and building UDTs), we aimed to investigate whether such update mechanisms are already being considered in practice. In addition, we explored how these mechanisms correlate with the data integration levels. This question was included based on the understanding that the level of integration adopted in the creation and use stages often determines the appropriate level of integration at the update phase. In some cases, the choice may even lead to a shift in the integration level later on, once the digital twin has already been established and/or utilized. The responses gathered from the survey reveal that strategies for updating UDTs are still evolving, with varying degrees of integration and automation. While some practitioners have clearly defined mechanisms, such as using Cesium Ion for cloud storage and APIs for real-time data updates, others have yet to formalize their update strategies, with some still relying on manual updates, simple front-end modifications, or retrieving data directly from their data owners. This highlights the diversity in maturity levels and approaches across UDT projects. Comparing data integration levels and update mechanisms leads us to confirm the link between the levels of integration defined during the creation and use phases and how they affect the update mechanisms. The integration levels determine the complexity and automation of the update strategies, creating distinct categories for UDT systems. For instance, at level 1, updates are primarily managed through versioning systems and ADEs, such as Scenario ADE. At level 2, ETL tools and predefined rules are employed to automate the data flow, ensuring that new data integrates smoothly with the established database.

At level 3, updates are frequently automated through the integration of real-time data APIs, such as those from weather stations, IoT devices, or urban sensors. These updates are then visualized on the front-end without requiring manual intervention. Additionally, practitioners may use external tools that allow for automated data ingestion, ensuring that new data is regularly pushed to the system. This level enables a dynamic and interactive digital twin, where updates happen in near real time, and the data continuously reflects the evolving state of the urban environment.

This work was conducted as part of a collaboration with **Dr. Binyu Lei** and **Prof. Filip Biljecki** in 2025, and the results are currently under preparation for publication together with additional relevant findings.

CHAPTER 4: Rethinking UDT Architectures: A Full- Stack CityJSON-Based Platform

Preface

Building on the conceptualization of the levels of integration introduced in the previous chapter, this chapter takes a step toward operationalization of Urban Digital Twins (UDTs) through real-world implementations. While the levels of integration framework provides a conceptual basis for reasoning about how data can be integrated into a UDT, its relevance and applicability can only be evaluated through concrete use-case requirements. In this context, this chapter focuses on the design and development of a full-stack UDT platform that serves both as a test environment for the proposed integration levels and as a practical contribution to ongoing UDT development. It primarily addresses the lack of a detailed UDT implementation methodology by deriving and comparing technical solutions, thereby providing the guidelines needed to build an operational UDT. By doing so, this work helps bridge a gap in the literature regarding UDT technical requirements and supports current development practices by leveraging lightweight standards and solutions.

A key motivation for this work is the current reliance of many UDT implementations on CityGML, where 3DCityDB is commonly used for storage and management. In parallel, CityJSON has emerged as a lightweight and developer-friendly encoding of the CityGML data model, designed to facilitate web-based workflows and software development. To better align the platform with these design principles, this chapter adopts the CityJSON Database (CJDB) as the backend storage solution. CJDB is designed to store and manage CityJSON files simply and efficiently, enabling direct access to geometries and attributes for querying, updating, and serving data to client applications. This choice also supports the direct visualization approach that will be described in this chapter, as CityJSON objects can be retrieved and delivered to the viewer without requiring intermediate tools.

In addition, the state of the art rarely details the minimum technical requirements for developing an operational UDT framework. This gap motivates the following research question: **RQ4. How can an Urban Digital Twin framework be developed to support different levels of data integration in practice?**

As a result, we developed and documented an open-source Urban Digital Twin platform, named City2Twin, designed to support data integration, storage, visualization, and analysis within a unified architecture. CJDB provides a practical foundation for implementing and evaluating an operational UDT platform based on lightweight standards and web-oriented tools. The platform adopts CityJSON as its core encoding for 3D city models, leveraging its lightweight structure and direct web compatibility, and supplements it with a modular architecture capable of handling both static and dynamic urban data. By combining open-source tools, standardized data models, and direct database-client interaction, the proposed platform challenges current UDT implementation paradigms and demonstrates an alternative scalable approach. This chapter details the technical choices made at each stage of the platform's development, including data preparation, database selection, visualization architecture, and dynamic data management. Through comparative experiments, the work highlights the advantages of separating static and dynamic data, maintaining direct interaction between system

components, and aligning architectural decisions with data integration requirements across the UDT lifecycle. Beyond presenting a single implementation, the chapter aims to extract transferable insights and technical requirements that can inform future Urban Digital Twin developments.

By the end of this chapter, the reader will gain a concrete understanding of how CityJSON can serve as a strong candidate for lightweight Urban Digital Twins, and how full-stack architecture can support flexible, maintainable, and interoperable UDT systems. The outcomes of this chapter therefore serve as a technical and methodological bridge between the conceptual framework introduced earlier and the full-stack implementation that will be detailed in chapter 5, where the proposed data integration levels are tested, and evaluated through the City2Twin platform.

Based on the article (Rafamatanantsoa and **Jeddoub** et al. 2024)

City2Twin: an open urban digital twin from data integration to visualization and analysis.

Rafamatanantsoa, B. P., Jeddoub, I., Yarroudh, A., Hajji, R., and Billen, R. ISPRS Archives, XLVIII-2/W8-2024, 387–394. doi:10.5194/isprs-archives-xxviii-2-w8-2024-387-2024 (2024).

Abstract: Urban Digital Twins have gained significant interest in the urban and geospatial fields, enabling interactive visualization and advanced analysis of cities across various domains. However, current implementation approaches are heterogeneous in terms of data and approaches. Furthermore, most implementations are based on specific needs. This project develops a comprehensive framework for Urban Digital Twins, focusing on data integration, storage, visualization, and analysis, all using open-source tools. Our approach integrates various data types, including 3D city models, dynamic air quality data, and external data imported from the client side, such as vector data, 3D city models, and point clouds. We conducted a series of experiments for each step and tackled various challenges. Many configurations are applied before integrating the 3D models, including ground reprojection, geometry type conversion, and format conversion. For data storage and management, we performed several comparative tests between 3DCityDB and CJDB, which led us to choose CJDB for its simplicity and lightweight nature. A client interface built with the Giro3D framework (based on Three.js) connects directly to the 3D model database via a Flask server. Dynamic data is retrieved via external APIs and stored in a separate database following the SensorThings API standard, allowing time-series analysis. Our framework is standardized and designed based on open-source software, emphasizing the openness, transferability, reusability, and maintainability of Urban Digital Twin. The **City2Twin** platform proposes significant improvements in data analysis, highlighting the importance of having a separate database for storing static and dynamic data, as well as the importance of direct interaction between the client interface and the 3D database for data updates and management. To the best of our knowledge, this work is the first Urban Digital Twin initiative that relies on the CityJSON format, which defines itself as more developer friendly.

Keywords: Urban Digital Twin, Data integration, 3D data visualization, 3D data analysis, CityJSON, Internet of Things (IoT)

4.1 Introduction

Urban Digital Twins (UDTs) are the new paradigm that shifts from traditional urban analytics to integrated, sustainable, and evolving methods and tools for city management and maintenance (Ketzler et al., 2020). Unlocking the potentialities of UDTs starts by defining the minimum technical requirements to implement an open, operational, and easy-to-use interface (Jeddoub et al., 2023). The shift toward a web developer-friendly approach and lightweight specifications and standards has led to addressing some of the UDT challenges related to data integration and interoperability, as well as serving users (especially cities) with integrated and dynamic urban platforms. Many questions about the suitable data models (Lehner et al., 2024), database systems (Kasprzyk et al., 2024), and visualization tools to create UDT platforms remain open to research (Gitahi and Kolbe, 2024). An examination of the literature reveals that most current implementations rely either on specific applications, such as energy use, mobility, and urban planning, etc., or on available data. Up to date, the state of the art is not well developed regarding how UDTs are implemented in practice and does not provide an in-depth analysis of the technical requirements. Furthermore, all attempts to build UDT are based on CityGML and there is no active initiative that relies on the CityJSON format, which defines itself as more developer friendly.

The aim of this chapter is to develop an open and standardized UDT framework that addresses current challenges of multi-source data integration and visualization in a lightweight format. The main steps involve the preparation and integration of the data sets in a standardized way, their storage using a database management system, as well as the visualization using open-source tools. The standardized framework is designed using an open software approach that emphasizes openness, transferability, reusability, and maintainability of UDT.

4.2 Related work

In recent years, several cities have taken the initiative to develop UDTs to enhance urban management. Jeddoub et al., (2023) established a comprehensive state-of-the-art review of current UDT implementations. Some of these projects are still in the prototype phase, others are under improvements, and some are already fully operational. The approaches for implementing UDTs generally depend on specific needs and the availability of potentially heterogeneous data. For instance, the EnSysLE project is used for building energy simulation and Amsterdam 3D focuses on city planning (T. Santhanavanich et al., 2022). As a result, various methods are developed based on different use cases. Most cities, like Helsinki 3D, Livable City Digital Twin or Berlin 3D, adopt web-based visualization, while others, such as Virtual Gothenburg use game engine visualizations. Among these implementations, some rely on commercial solutions, such as 3D ArcGIS online for karlskrona city, rather than open-source alternatives, which can be costly and offer limited flexibility and customization. Several open-source tools are available for visualizing UDTs on both web and desktop platforms. Most of these viewers rely on CesiumJS with the 3D Tiles data format, such as the Urban Energy Dashboard (Würstle et al., 2020) and the Digital Twin of Sofia City (Dimitrov and Petrova-Antonova, 2021). Other viewers utilize the Three.js library, including Piero, Ninja, and 3D Bag Viewer.

The use of different standards ensures data compliance and interoperability. The CityGML standard defines a conceptual model and an exchange format for the representation, storage, and sharing of virtual 3D city models. It can be encoded in several formats, including GML, which is considered verbose and heavy. An alternative encoding is CityJSON, a JSON-based format that is lighter and easier to understand. Due to its simplicity, several tools and programs have been developed to create, analyze, visualize, and edit CityJSON files, such as Ninja, 3dfier, and Cjio. Another method for storing the 3D model is database encoding, either in a relational database such as 3D City Database 3DCityDB, the most widely used, or CityJSON Database CJDB (Powałka et al., 2023) for CityJSON-based storage, or in a NoSQL database like MongoDB (Kasprzyk et al., 2024). These databases not only store the 3D model but also enable efficient management.

Dynamic data is an essential component that enriches the UDTs, and there are several ways to store and manage these data. For example, dynamic data can be integrated via a Dynamizer extension for CityJSON-based visualization (Boumhidi et al., 2024), or through the Dynamizer module, which ensures the direct integration and representation of urban object properties through time-series data in CityGML 3.0 (Kutzner et al., 2020). This data can also be stored in a database following the SensorThings API (STA) standard, which manages not only the data itself but also the various elements related to data flows (Santhanavanich and Coors, 2021).

4.3 Method

This section provides a detailed overview of the methodology summarized in Figure 4.1. We have proposed a generic methodology, supported by a series of technological considerations. This methodology is divided into three major steps: (1) **data preparation and integration**; (2) **data storage and management**; and (3) **data visualization and analysis**. For each step, multiple tests were conducted to ensure the selection of the most suitable tools and approaches for optimal results.

4.3.1 Study Area and Datasets

The data used consists of two different types: static data and dynamic data. The static data represents the 3D models of urban objects according to the CityGML standard and encoded in JSON format (i.e., CityJSON). This 3D model covers the Outremeuse district in the municipality of Liège, Belgium. The district has an area of 1.247 km² and a population of 4,470 inhabitants, corresponding to a population density of 3,586 inhabitants per km² (Source: JRC - Joint Research Centre of the European Commission, 2015).

The 3D city model contains urban objects such as buildings, vegetation, roads, and city furniture (See Table 4.1). This model is the outcome of the previous project introduced in (Ballouch et al., 2024). The CityJSON file is generated using Geoflow²⁵ for building reconstruction. The vegetation is generated at LoD2 following a three-step process (i.e., classification, segmentation, and modeling). Roads are reconstructed using an FME workbench and the lamppost are the results of an automatic detection and 3D modeling pipelines of city furniture objects using a camera LiDAR fusion approach, while

²⁵ <https://github.com/geoflow3d/geoflow-bundle>

integrating KPConv for semantic segmentation and connected components for instance segmentation (Doi et al., 2024).

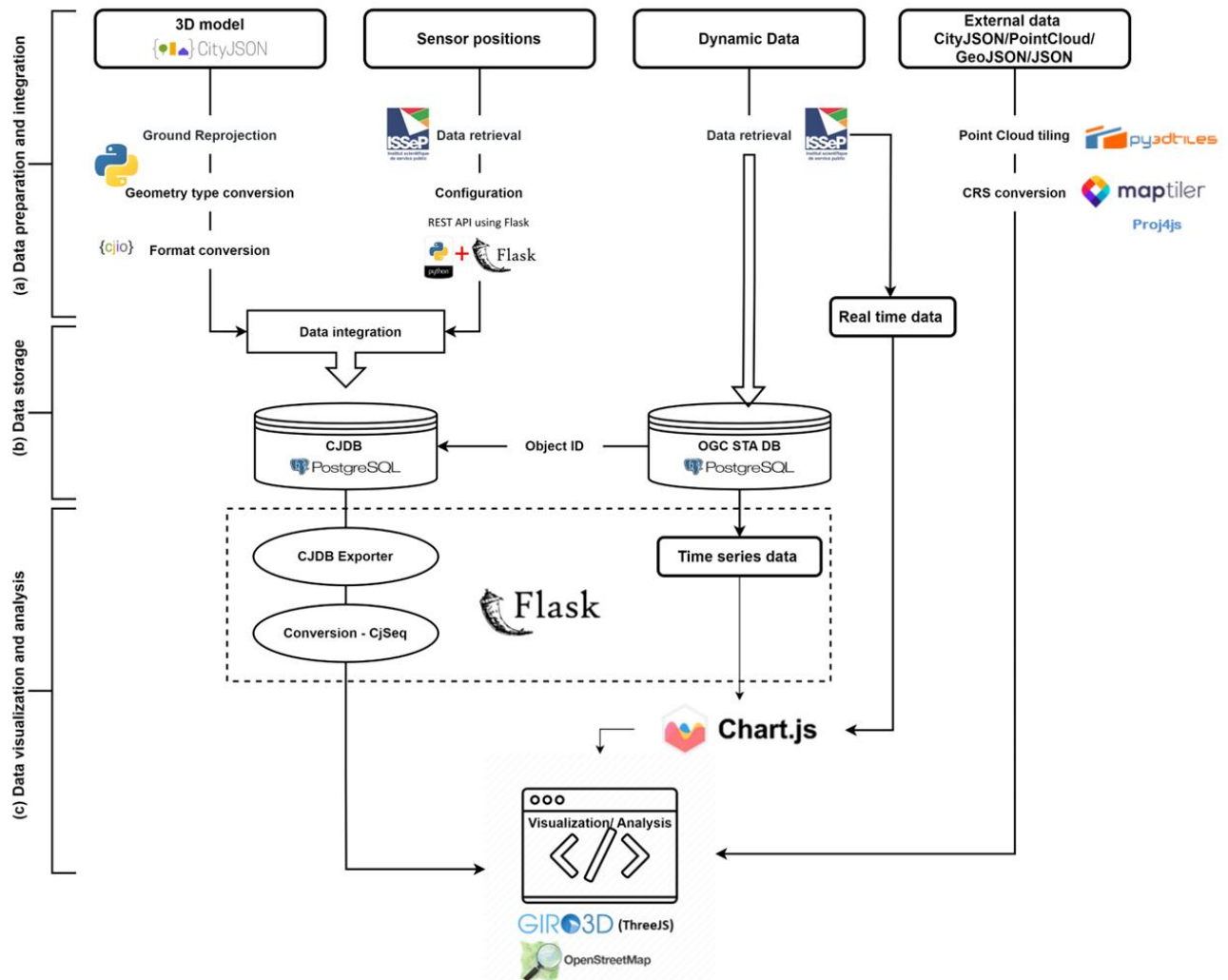


Figure 4.1. City2Twin methodology overview

Table 4.1. 3D city model of Outremeuse characteristics.

Object Type	Count	LoD	Geometry Type	CRS
Building	3901	2.2	-	-
BuildingPart	3815		<i>Solid</i>	3812
Vegetation	1476	2.0	<i>MultiSurface</i>	3812
Road	341	1.0	<i>Solid</i>	3812
City Furniture (StreetLights)	178	2	<i>Geometry Instance</i>	31370

The dynamic data used in this project represents the air quality data of the city of Liège which is provided by the [ISSeP](#) and retrieved via APIs. The APIs provided in a JSON format, contains date and time of measurements, concentrations of each pollutant in $\mu\text{g}/\text{m}^3$, temperature in $^{\circ}\text{C}$ and characteristics and position of each sensor (see **Figure 4.2**).

Monitoring stations are distributed across the city (refer to **Figure 4.3**). They are measured by sensors placed on lamp posts distributed across the city (see **Figure 4.4**). Air quality is determined by the concentration of pollutants in the atmosphere. There are several pollutants in the air, with concentrations measured in micrograms per cubic meter ($\mu\text{g}/\text{m}^3$). Among these pollutants are fine particles with a diameter of less than 2.5 microns and 10 microns (PM2.5 and PM10), Ozone (O3), Nitrogen dioxide (NO2), and Nitric oxide (NO). To calculate the overall BelAQI Air Quality Index, the sub-indices for each pollutant are determined from the graded scale in **Figure 4.5**. The highest sub-index determines the overall air quality index.



Figure 4.2. Retrieved API responses for sensor ID = 42: sensor metadata and location (left) and air quality observations (right).

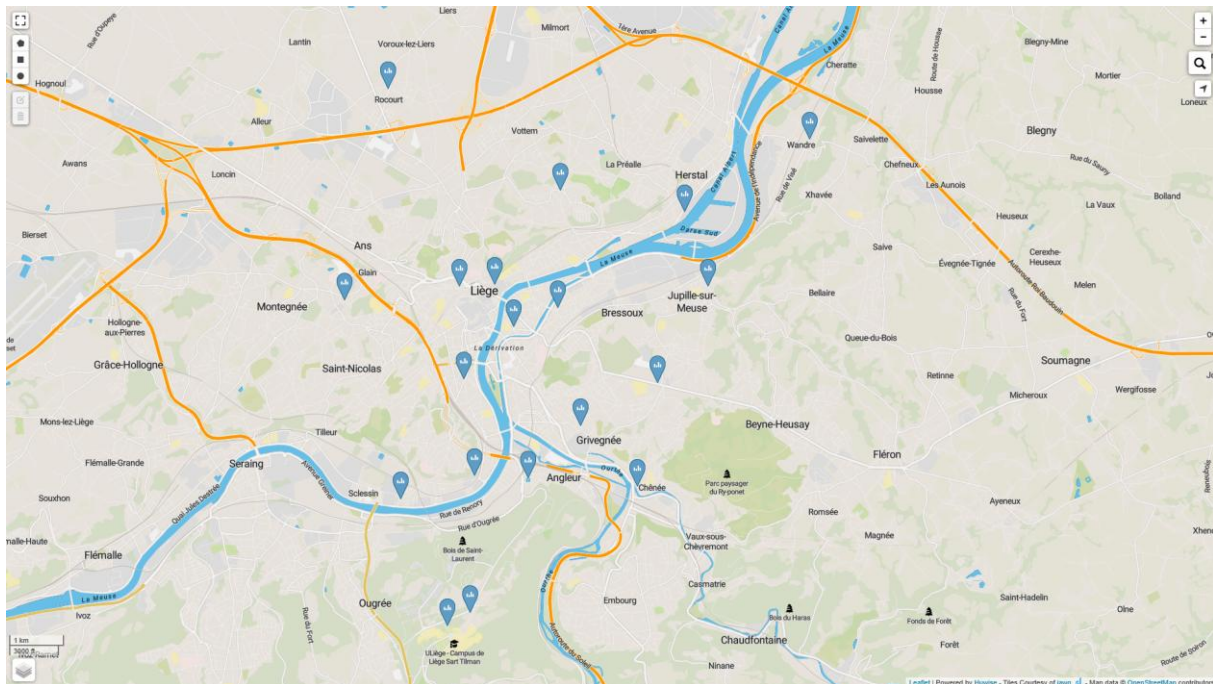


Figure 4.3. Distribution of air quality monitoring stations in the city of Liège (Source: OpenDataLiège).

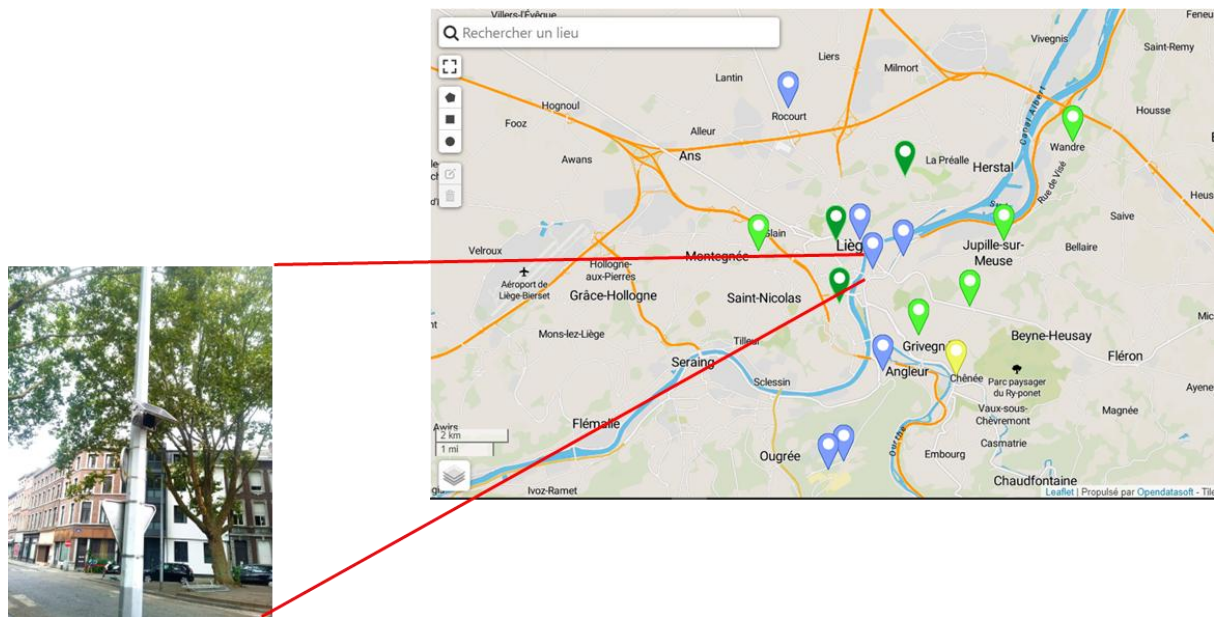


Figure 4.4. Air quality monitoring station in the OutreMeuse district (Study area).

Index	Classification	PM ₁₀ hour	PM _{2.5} hour	O ₃ hour	NO ₂ hour	
1	Excellent	0 - 11	0 – 3.5	0 - 30	0 - 10	
2	Very good	11 - 20	3.6 – 7.5	31 - 65	11 - 15	
3	Good	21 - 35	7.6 – 10	66 - 75	16 - 20	
4	Fairly good	36 - 45	11 - 15	76 - 90	21 - 30	
5	Moderate	46 - 60	16 - 20	91 - 110	31 - 40	
6	Poor	61 - 80	21 - 35	111 - 150	41 - 45	
7	Very poor	81 - 95	36 - 50	151 - 180	46 - 50	
8	Bad	96 - 110	51 - 60	181 - 210	51 – 60	> O ₃ information threshold
9	Very bad	111 - 140	61 - 75	211 - 240	61 - 75	
10	Horrible	>140	>75	>240	>75	> O ₃ alert threshold

Figure 4.5. BelAQI hourly index classes based on the relationship between daily maximum hourly concentrations and daily average concentrations (maximum 8 hours for O₃). (Source: irceline.be)

4.3.2 Configuration of the workspace

A development environment was configured to ensure modularity, reproducibility, and compatibility between the different software components. A dedicated Python 3.10 virtual environment was used to isolate backend dependencies from the host system, thereby preventing conflicts with other installed packages or Python versions.

All backend libraries were installed within this environment. The backend architecture relies on a Flask-based server responsible for handling HTTP requests and exposing RESTful endpoints used by the frontend client. In parallel, a dedicated development server was employed for the frontend to support iterative testing and visualization. Frontend dependencies include visualization and rendering libraries such as Three.js. This separation between frontend and backend environments enables independent development, testing, and debugging of each component.

Detailed installation commands and configuration scripts are provided in APPENDIX B to support reproducibility.

4.3.3 Data Preparation and Integration

This is a necessary step in our approach, given the heterogeneous nature and sources of the data, which include geodata and sensor data. This step involves carrying out all the necessary preprocessing and configuration.

Static Data Integration

This framework aims to support the integration of various data types, such as 3D city models, vector data, and 3D point clouds. Before integrating this data, it must comply with the different tools and methods used. All urban objects in the CityJSON data, such as buildings, trees, and roads, were projected onto the ground for two reasons. First, the surface of base maps in web viewers has a Z value of 0; however, our data includes elevation values, causing objects to float above the ground surface. We attempted to import a Digital Elevation Model (DEM) from online sources, but it does not perfectly match the actual terrain variation. As a result, some buildings appear to float while others sink into the ground. Another reason for this process is that roads, which are not projected onto the ground and therefore are not flat, cannot be imported into the CJDB database (as this is the chosen tool for storage, refer to section 4.4). We assumed that the CJDB import functionality requires flat geometry to populate the *ground_geometry* column, whereas our road objects do not have a flat geometry. To achieve the reprojection, a Python script was developed to determine the minimum Z value for each object and subtracted this value from all other Z values in the same object.

The 3D objects of the City Furniture type (lamp posts) have a geometry type called Geometry Instance. This type of geometry is difficult to parse and load with our visualization tool. Therefore, it has been converted to a MultiSurface type using the CJDB export, which automatically converts geometries of the Geometry Instance type into a MultiSurface type during the export.

The database used in this project does not directly support CityJSON files. They must be converted into a CityJSON Text Sequence format with a «.city.jsonl» or «.jsonl» extension. This allows for the potentially large CityJSON file to be decomposed into individual

entities. This conversion was performed using *cjio*, a Python CLI to process and manipulate CityJSON files.

The measurement stations on which the sensors are placed are also stored as city objects in the 3D model database. Only the stations in our study area are represented by 3D city model (lamppost), while those outside the area are represented as points. However, they must conform to the 3D database schema (i.e., adhere to the column type and data structure requirements).

Dynamic Data Integration

Dynamic data is retrieved via external APIs. Since this data may vary over time, it is not optimal to store it directly in the model to avoid overloading the 3D model. Thus, the dynamic data is stored in a database following the SensorThings API standard. This database enables not only storage, but also better management of various elements related to *datastreams*, such as the description of the sensors used, the observed properties. Before recording the data, the entities shown in **Figure 4.6** need to be created. For each measurement station, it must be associated with a *Thing* that represents the station, along with its *Location*, the *ObservedProperty* corresponding to different pollutants and temperature, the *Sensor* measuring the observed properties, and *Datastream* that links the sensor used and the observed property for each *Thing*. To optimize storage, different properties measured by the same sensor (such as the case for PM10 and PM2.5) are grouped into a *MultiDatastream*. In this work, we focused on the measurement station in our study area, and the entities were created only for this station. However, it can be easily expanded to include all stations.

Creating these entities in the STA database can be accomplished using the FROST server, a server implementation of the OGC STA, which is deployed using Tomcat. The server is linked to the STA database to execute all queries for creating these entities. Observations can also be recorded via the Frost server; however, this can complicate the process since another server must be maintained. Therefore, we wrote an SQL script on our backend server that executes a query every 5 minutes to record the observations in the database.

4.3.4 External Data Integration

External data can be directly imported into the platform without being stored in a database, allowing users to work with their own data without the constraints of database permissions. They can import CityJSON files, GeoJSON files, or 3D Point Clouds from their local machine. For GeoJSON files, the visualization tool supports the import of this type of data. However, there is no default support for CityJSON, so to parse and load the CityJSON data into the Three.js scene (the visualization tool adopted in this work see section 4.5), a library called *cityjson-threejs-loader* is used. Similarly, the visualization tool does not allow for direct visualization of point cloud files in .LAS format; therefore, it must be converted into a 3D Tiles point cloud using the Python library *py3dtiles*.

Since the data may have a different Coordinate Reference System (CRS) than the default one (EPSG:3812), reprojection must be performed; however, this conversion is not applied to the entire 3D model but rather to the transformation matrix that positions the object correctly in the scene. The conversion can be done using libraries such as *MapTiler* or *proj4js*.

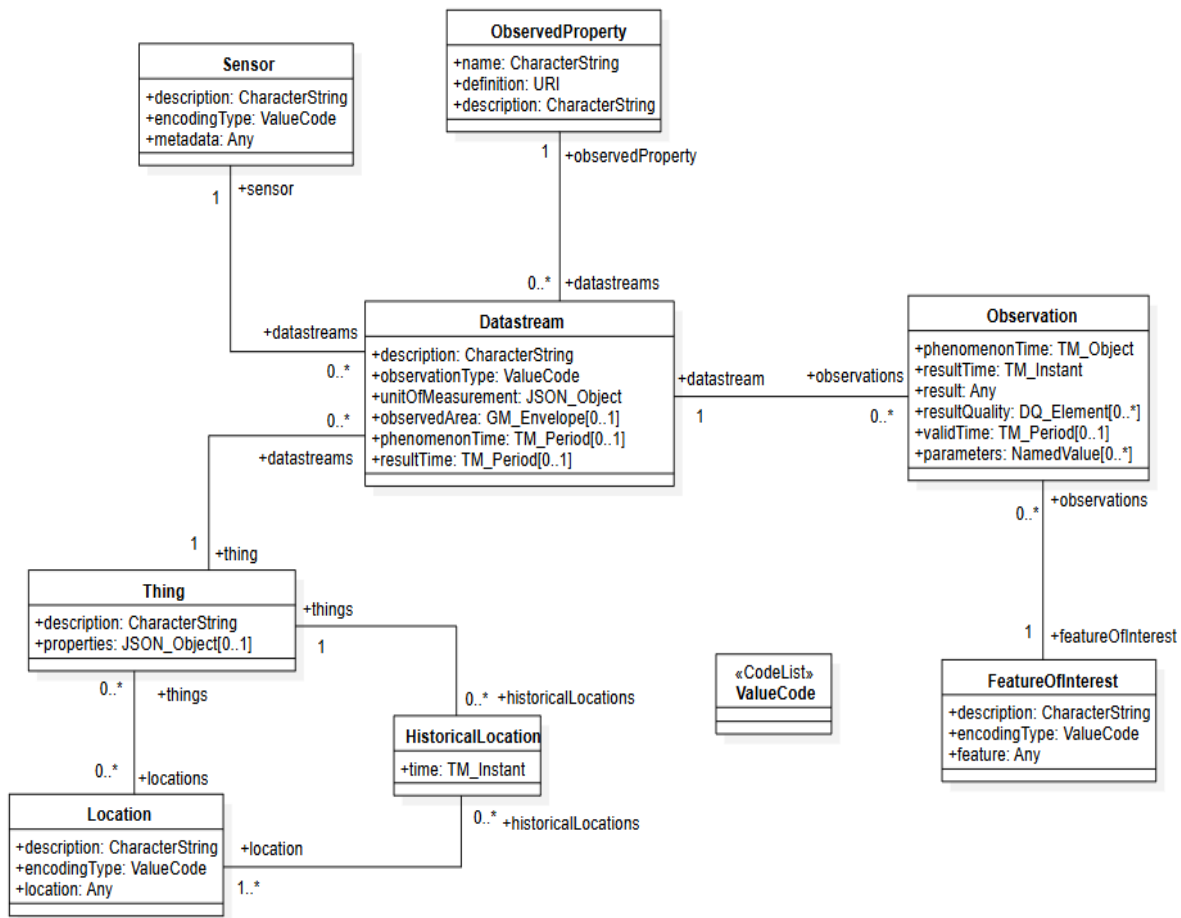


Figure 4.6. UML diagram of the STA data model.

4.4 Data Storage and Management

We used two separate databases: one for static data and one for dynamic data to effectively store and manage them. Both databases are interrelated. Both data models (CJDB and STA DB) are implemented in PostgreSQL. This facilitates the management of both databases by allowing them to be manipulated from the same interface, pgAdmin.

4.4.1 Selection of the 3D Database

Since there are several approaches to storing 3D data in a database, a choice had to be made. Several criteria were considered, including simplicity, query speed, storage volume, and support for geospatial functionalities. We had to choose between two databases that are most suitable for our project: 3DCityDB and CJDB. Various tests were conducted, allowing to establish the comparison.

To better analyze the performance between the two databases in terms of query speed and storage, we conducted tests (similar to those conducted by (Powalka et al., 2023)) using our own data. We maintained the same dataset for both databases, which consists of a 3D model in CityJSON format containing nearly 9,500 urban objects.

Once the model is imported into both databases, we already notice a difference in the size of the databases, as presented in Table 4.2. This can be explained by the complexity of the

3DCityDB schema, which stores the different elements of the 3D model in separate tables. 3DCityDB contains 66 tables, in contrast to CJDB, which compresses the model and contains only 3 tables. In CJDB, the geometries and attributes are stored in a single table called *city_object*. The 3D geometries are stored in a column of JSONB type.

To analyze the query speed on both databases, we selected several queries for comparison. The first three queries are applied to attributes, while the fourth one focuses on geometry. Although the queries are not exactly the same due to differences in the database schemas, they return similar results corresponding to our requests. Each query is performed 10 times, and the average results were taken. The time results of these queries are presented in Table 4.2.

Table 4.2. Comparison of database sizes and query times for CJDB and 3DCityDB.

Criteria	3DCityDB v4.4.0	CJDB v2.1.0
DB volume (MB)	252	97
Q1: Retrieve a building object using its ID (ms)	33.36	33.54
Q2: Add a new attribute to objects of type «Building» (ms)	53.75	76.87
Q3: Retrieve buildings with LoD 2 (ms)	191	302
Q4: Retrieve geometries of 100 buildings in 3D GeoJSON format (ms)	92	48

After conducting several tests and comparisons between the two databases, we can confirm the results of the test conducted by (Powałka et al., 2023), which shows the maturity of 3DCityDB in handling semantic surfaces and attributes (Q1, Q2 and Q3) consistently. However, the main advantage of using CJDB lies in its simplicity of use and understanding, as well as its method of storing 3D geometries, which makes them easy and rapid to retrieve (Q4), and results in smaller sizes in the database. In addition, CJDB is the most suitable database solution for the CityJSON-based encoding with the direct approach of visualization presented in section 4.4.1.

We used Postman to perform the test of Q4 for 3800 buildings, the results are presented in Table 4.3. The difference in response size is due to the number of decimal places. 3DCityDB limits the maximum number of decimals to 9 for longitude and latitude and 3 for the Z coordinate. In contrast, CJDB imports all decimal places, up to 15 digits for latitude, longitude, and Z coordinates in our case. Therefore, optimization is required for CJDB. To address this issue, a line of code was added to the same script used for CRS conversion to round the coordinate values. To ensure millimeter accuracy, spherical coordinates are rounded to 8 decimal places, and since the Z coordinate is in meters, 3 decimal places are sufficient. The result of this optimization is presented below. We also analyzed its impact on database size. Regarding database size, it became about 1/3 lighter than before. Additionally, the difference in response size is significant, as shown in Table 4.4.

Following the conducted tests, the difference in query response times for attributes is not significant at the scale of our data and our purpose. Furthermore, given that most current UDT implementations are generally based on the 3DCityDB, we found it interesting to

further explore the performance and limitations of CJDB. Several optimizations have been applied to the database to enhance its performance (see Section 4.4.3).

Table 4.3. Comparison of size and query time for 3800 buildings.

	3DCityDB v4.4.0	CJDB v2.1.0
Response time	4.068 sec	2.423 sec
Response size	24.11 MB	32.09 MB

Table 4.4. Comparison between CJDB database size and response time before and after rounding.

	Before rounding	After rounding
Response Time	2.42 sec	2 sec
Response Size	32.09 MB	19.58 MB
Total Database Size	97 MB	33 MB

4.4.2 Importing static model to CJDB

To import the data to the CJDB, this step involves converting CityJSON to CityJSON Text Sequence format. The conversion from .json to .jsonl format was carried out using the `cjio` library and then the data was imported into the database following the instructions in the official GitHub of CJDB²⁶. If multiple files need to be imported into the same database, they must have the same CRS in the database. In our case, the two CityJSON files do not have the same CRS, so we need to add the `--transform` option for importing the second file, which transforms the CRS of the input file to the CRS of the existing schema.

4.4.3 Optimization of the CJDB Database

According to the CJDB database schema, the footprints of each object are stored in a PostGIS geometry-type column, called *ground_geometry* (refer to **Figure 4.7**). During data import, the CJDB importer retrieves the ground geometry of each object; however, in some cases, this function is not properly executed. For example, for some objects in our CityJSON data, the column *ground_geometry* contains only an empty Multipolygon instead of a Multipolygon with geometries.

²⁶ <https://github.com/leoleonsio/cjdb>

We assumed that this issue occurs because, during data import, the CJDB importer tries to detect a flat ground surface, meaning that the Z values of the polygons are identical or have minimal differences. If this is not the case, it returns an empty geometry.

To ensure that the footprints are always stored, regardless of the case, a Python script was written and should be executed once the data has been properly imported into the database. The script retrieves the ground geometries of each object and adds them to the *ground_geometry* column.

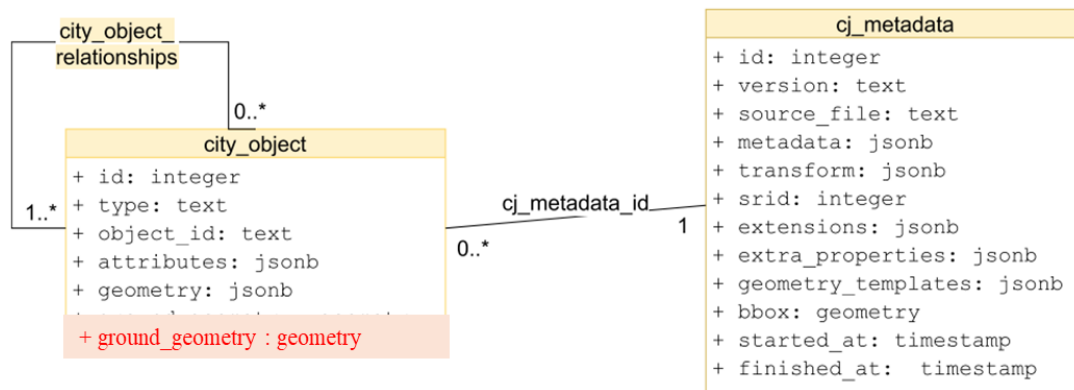


Figure 4.7. Optimization of the CJDB database *ground geometry*.

4.4.4 Integration of dynamic data

The dynamic data is stored in STA database. Before recording data, all the STA instances need to be created. Each measurement station needs to be associated with a *Thing* instance, *Things locations*, and *datastreams* that link the *sensor* and the *observed_property*.

Table 4.5 summarizes the mapping of our station to the STA data model.

Table 4.5. Creation of STA entities and their mapping to the case study measurement station.

STA entities	Mapping to our measurement station
Thing	Representing the object on which the sensor is mounted, the lamppost.
Locations	Positions of measurements station.
ObservedProperty	6 observed properties corresponding to the following measurements: NO2, NO, O3, PM10, PM25 and temperature.
Sensor	One sensor measures PM2.5 and PM10, while a separate sensor is used for each of the other observed properties.
Datastream	A datastream is created for each combination of ObservedProperty and Thing. This can be an issue for data storage since an observation is generated for each datastream. Observed properties measured by the same « sensor » on the same « Thing » are therefore grouped into a MultiDatastream.
MultiDatastream	PM25 and PM10 properties are grouped into one MultiDatastream.

Creating these instances is done using the FROST server by submitting the queries for each entity at the following URL: <http://localhost:8085/FROST-Server.MQTTTP-1.13.0/v1.1>. This URL indicates the location of the FROST-Server and the version of the API to use, which is hosted on the Tomcat server at <http://localhost:8085>.

In our case, we focused on the Outremeuse sensor, ID=42, in which we have the 4 datastreams (NO, NO2, O3 and temperature) and one Multi_datastream (for PM10, PM25). Once the STA instances are created, we need to record the observation in the STA DB. For this, we retrieved the data every 5 minutes via the external API (lastdata). A server-side SQL script is written to record the data. Since the frequency of new data publication varies, ranging from 10 to 20 minutes and sometimes extending to several hours, data recording is only performed when there is a change in the data. Indeed, we recorded the observations directly from the Flask server by connecting the database to our server using *psycopg2*. An SQL query is then executed to store the data in the database. For this method, data is retrieved using the `requests.get()` function, then converted into JSON format. The data is then grouped to comply with the database schema and the instances that were created. For example, PM2.5 and PM10 values are grouped into one list. All observations are then combined into a list of structured objects. The `cursor.executemany(query, data)` method is used to insert multiple rows into a database table in a single operation, which is more efficient than executing individual insert queries. Recording the observations is performed periodically using a dedicated thread. The thread, which is started in the background when the application is launched, calls a function that retrieves data every 5 minutes using `time.sleep(300)`.

4.4.5 CityThings concept

After integrating the data into their respective databases, linking both databases relies on CityThings concept. The concept was originally implemented to link the SensorThings and CityGML standards by establishing a link between the sensor and the 3D city model element to which the sensor observations relate (Santhanavanich and Coors, 2021). We extended this concept based on CityGML and 3DCityDB with lightweight encodings (e.g., CityJSON and CJDB). It allows linking static and dynamic databases, where an urban object (lamp post stored in CJDB) corresponds to an object where a sensor is placed (measurement station stored in STA Database as a *Thing*). Technically, to ensure this link, the instance of *Thing* defined in the UML diagram of the STA data model stores as a property the unique identifier of its corresponding 3D urban object (which, in our case, is the ID of the lamp post). **Figure 4.8** shows an example of how the urban object identifier is stored in the *Thing* entity.

By using this approach, we can retrieve information about the 3D urban object corresponding to the sensor data flows, and vice versa (see **Figure 4.9**).

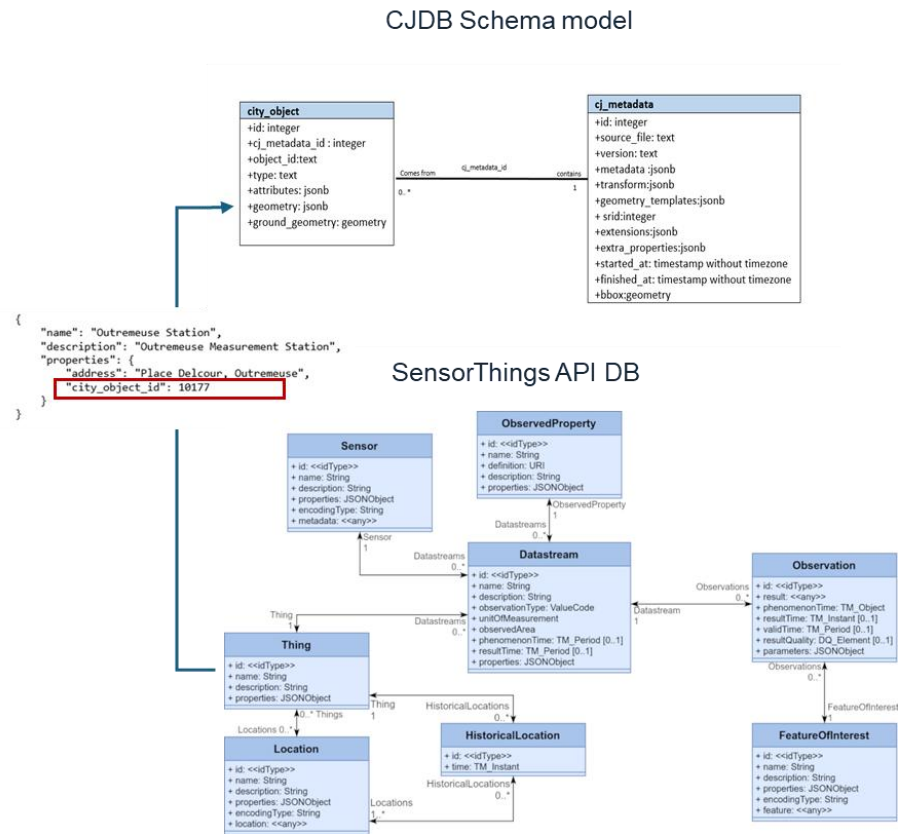


Figure 4.8. Linking CJDB and STA Databases following the CityThings Concept.



Figure 4.9. The CityThings concept applied to a sensor in the study area: (a) Static attributes from the modeling process, (b) dynamic attributes from the STA DB.

4.5 Data Visualization

4.5.1 Choice of visualization tool and approach

This step involves retrieving objects from the database and visualizing them on web-based platforms. We have selected web visualization as it is more optimal and does not require any additional software or hardware installation by the user. At this stage, we have compared two existing approaches in literature considering various challenges related to the model size and loading time. For example, the retrieval of geometries and their visualization in a web-based approach follows two approaches.

- **Direct approach:** the objects are retrieved directly from the database by running an SQL query. The returned object is in GeoJSON or CityJSON format, including both attributes and geometries. In this approach, there is a direct connection between the client and the database. Updates are made automatically after any modification, such as the deletion or modification of urban objects. However, loading time and visualization can be quite slow and require more resources, as the data is loaded in a format that is not suitable for large data visualization.
- **Indirect approach:** is the most widely adopted in most UDT implementations. It consists of exporting manually 3D models and converting the model to 3D tiles format, storing them on cloud servers such as Cesium Ion and retrieving them by ID. The use of these visualization and streaming formats guarantees rapid visualization. However, a direct link between the client and the database where the objects are stored and managed is not assured. This means that any changes to urban objects require a new export for visualization.

Given that the study area is a neighborhood containing approximately 10,000 3D objects, the scale is relatively small. Moreover, this project aims to implement a solution that is not restricted by limitations such as the lack of a direct link between the database and the client or the need for manual data export and conversion into a format suitable for web streaming. It also aims to further explore the use of a database in the backend for the management of 3D objects within the context of UDT. Consequently, the direct approach was adopted.

Common UDT visualization tools and frameworks have been implemented and fully tested. Their differences lie in loading speed, supported formats, built-in features, and other aspects. We investigated two JavaScript libraries, CesiumJS and Three.js, to parse and load data in an interactive way. These are both web-based graphics libraries used to create interactive 3D graphics in a browser. CesiumJS specializes in geospatial applications and offers advanced geospatial features. However, with the direct approach, 3D data will be retrieved and viewed in 3D GeoJSON format to facilitate data flow from the database, through the server, and to the client. This method is very slow and requires more memory. For Three.js, while it lacks built-in support for geospatial functionalities, it can efficiently visualize data in CityJSON format in a shorter time by using the *cityjson-threejs-loader* library. Since we conducted our first attempt with the approach based on CesiumJS and 3D GeoJSON, we observed that the approach based on Three.js and CityJSON brings significant improvements in loading speed, memory usage, and navigation fluidity.

To address the lack of geospatial functionalities in Three.js, we used the Giro3D framework. It is a recent open-source JavaScript framework designed to visualize and interact with heterogeneous data in 2D, 2.5D, or 3D. Although it is based on Three.js, it integrates support for geospatial functionalities by using the Openlayers library. **Table 4.6** highlights the findings based on the tests conducted in this work.

Table 4.6. Comparative Analysis between JavaScript frameworks: CesiumJS, ThreeJS, and Giro3D.

	CesiumJS	ThreeJS	Giro3D
Data Formats Used	GeoJSON 3D	CityJSON	CityJSON
Visualization Speed	Slower	Faster	Faster
Memory	Large (~ 3 GB)	Light (< 1 GB)	Light (< 1 GB)
Geospatial Features	Advanced	Less Advanced	Advanced
Visualization parameters	Less Advanced	More advanced	More advanced

Although we initially tested the direct approach using CesiumJS, we switched to the Giro3D framework. This is due to several challenges, namely the retrieved 3D GeoJSON required additional preprocessing (i.e., closing open surfaces and adjusting the nesting of Solid geometries) before it could be rendered in Cesium. In addition, the large number of attributes associated with each object often require joins with parent entities, which make the retrieval and visualization inefficient, leading to longer query response times and slower scene loading. While partial optimization was achieved by separating geometry loading from attribute retrieval (attributes fetched on demand upon user interaction), these limitations motivated the transition to the Giro3D framework, which is described in detail in the following sections.

4.5.2 3D Data Retrieval based on Giro3D framework

Initially, when the user loads the page, an HTTP request is immediately sent to the server, which is a Flask server, that retrieves 3D data from the CJDB database. This database automatically exports data in CityJSON Lines format using the CJDB Exporter function. Once the data is retrieved, it must be converted to CityJSON format to be readable by the visualization tool. This conversion can be done using CjSeq, a program for creating, processing, and modifying CityJSONSeq files, as well as converting them to CityJSON files (Ledoux et al., 2024). The CityJSON data is then sent to the client side to be parsed and loaded into the Giro3D scene.

Figure 4.10 illustrates the UML sequence diagram of this architecture.

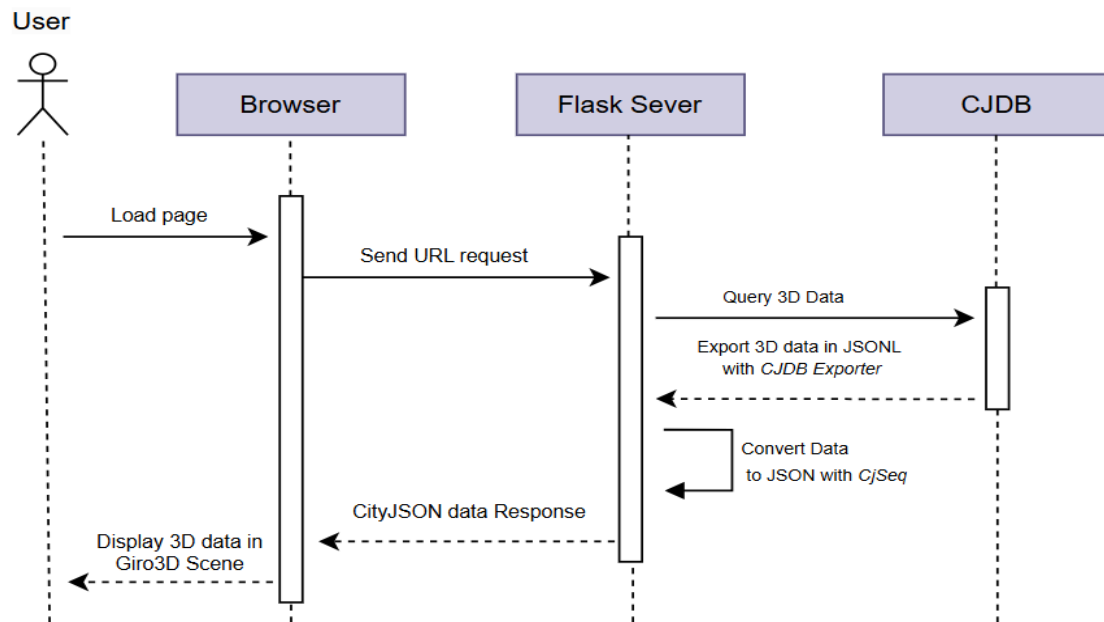


Figure 4.10. UML sequence diagram of 3D Data Loading.

4.5.3 Dynamic Data Visualization

The approach used for visualizing dynamic data is independent of the one applied to static data. This data is retrieved through APIs and can either be directly displayed on the interface or saved in the STA database for time series analysis.

Displaying Punctual Data

First, a request is made to fetch information about available sensors, such as their 2D position, ID, address, etc. This initial request allows the sensors to be displayed in the scene and enables modifications to the database if new sensors are added or removed. The second request is sent when the user clicks on a specific sensor. This request retrieves the latest measurements taken by the sensor, including the concentration of each pollutant, temperature, air quality index (AQI) value, and the timestamp of the measurement (aqi_timestamp). **Figure 4.11** illustrates an example of the results from the second request, which are then displayed in a graph using the Chart.js JavaScript library.

```

{
  "total_count": 1,
  "results": [
    {
      "aqi_value": 5,
      "indice_aqi": "5-Moyen",
      "aqi_timestamp": "2024-07-31T12:00:00+00:00",
      "no_rf": 1.842999999999997,
      "o3_rf": 17.875999999999999,
      "pm10_rf": 13.748333333333333,
      "pm25": 8.786,
      "no2_rf": 33.991333333333344,
      "bme_t": 23.87
    }
  ]
}
  
```

Figure 4.11. Example of retrieved data for a specific sensor.

Displaying time series data from the database

The time series values are retrieved from the STA database. An SQL query is executed to extract data from the *OBSERVATIONS* table, ordering the results by date and time. The results are then processed to group the data by date. Each date is associated with a dictionary containing values for various pollutant concentrations, temperature, and a calculated air quality index value (since this value is not stored in the STA database and must be calculated based on pollutant concentrations). The data is then formatted into a JSON dictionary, where the keys represent the date and time. These data are then returned and displayed in a graph. The sequence diagram of the dynamic data analysis is described in Figure 4.12.

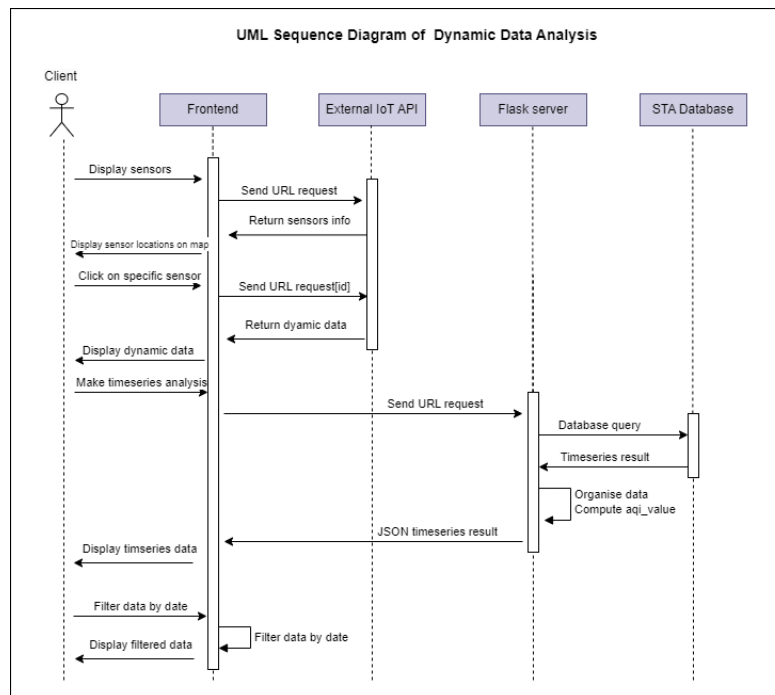


Figure 4.12. Sequence Diagram for dynamic data analysis.

4.6 Various features integrated into the platform

To provide a fully featured and user-friendly platform, a variety of functions have been implemented.

4.6.1 Retrieving information related to an object by its ID

By using the raycasting principle, we can retrieve information about the clicked object such as their *object_id*, the attributes of this object and the ID of the semantic surface if it exists. A div overlay is then displayed, containing this information (refer to **Figure 4.13**).



Figure 4.13. Retrieving information about a specific object (i.e., building).

4.6.2 Retrieving information related to a specific surface

If the model contains information about surfaces (e.g., wall surfaces, roof surfaces, or other attributes), this information can be retrieved using the surface ID obtained through the raycasting (see **Figure 4.14**).

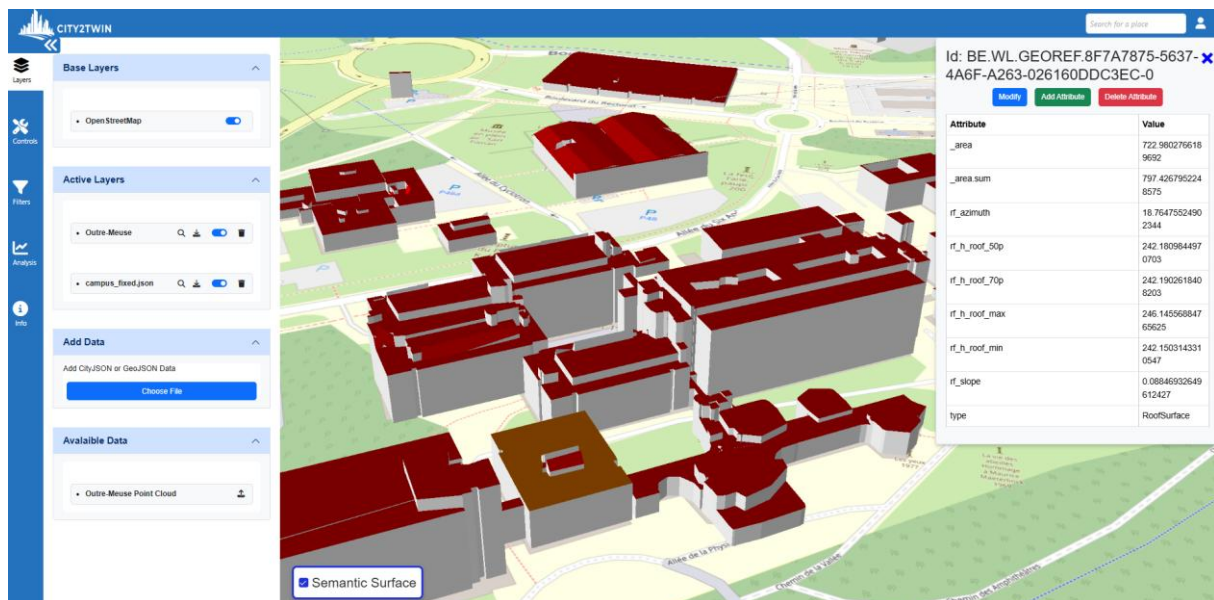


Figure 4.14. Retrieving information about a specific surface (i.e., Roof Surface).

4.6.3 Multi-layer management

A layer management section has been added to handle multiple data sets loaded in the scene. For each layer, we can toggle its visibility, zoom in on the layer, or download the layer for use in other applications. The layer management has base layer which is the

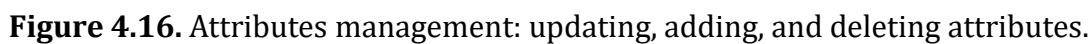
OpenStreetMap and the active layers are the layers imported in the client side or the one retrieved from the CJDB database (refer to **Figure 4.15**).



Figure 4.15. Managing multiple layers from various sources.

4.6.4 Attributes manipulation

The *div overlay* that appears after double-clicking on a specific object contains three buttons for modifying, adding, and deleting attributes. The process for these three functionalities is similar. After identifying the concerned 3D object, a filter is applied using its ID. Operations are then performed on this object: either assigning a new value to an existing attribute, adding a new attribute, or deleting an attribute (see **Figure 4.16**). To update the database, endpoints are created to execute an SQL query that modifies the database. If the attributes concern semantic surfaces, the modification approach differs. In this case, we modify the "semantics" property instead of the "attributes" property. Following the standard specifications, if we want to modify the properties of a specific surface, but several surfaces refer to a common property in the "surfaces" list, it is necessary to create a new object in the surface properties list. Then, we must update the surface value in the "values" list to refer to this newly added property. This procedure is also considered when updating the database.



For filters based on attributes, we specify first which layer the filter will be applied to. After setting various conditions based on the object type and attributes (see Figure 4.17-a), these conditions are grouped and then applied to the current city model to create and load in the scene a new city model containing only the desired objects.

For spatial filters, this functionality has not yet been integrated on the client side. It must be executed on the server side of the database using PostgreSQL spatial extension PostGIS and the `ground_geometry` column in CJDB, which contains the objects footprints in PostGIS geometry format. First, we specify the type of object, the type of spatial functions (DWITHIN, DWITHOUT, INTERSECT, or DISJOINT), and the type of geometry to be drawn directly on the base map (see Figure 4.17-b). These parameters are then sent to the server, and a spatial query is executed in the database, which returns the filtered objects.



Figure 4.17. a) Attribute filter: buildings of type «Building», trees < 20m, and roads for national use; b) Spatial filter: buildings located within 10m around a road segment.

4.6.6 Optimization of visualization

All visualization and control parameters can be adjusted based on the user's needs using the Dat.GUI library. The user can change the sky color, adjust light intensity, display objects based on the Level of Detail (LoD), and change the colors of urban objects based on a specific attribute. Additionally, they can customize the colors according to the type of object (e.g., Building, Road, etc.) or the semantics of the building (e.g., RoofSurface, WallSurface, etc.).

4.6.7 User management

To manage different permissions for modifying data in the database, a user management system has been integrated. A separate database is used to store user information, including name, email address, password, and role, which can either be *admin* or *user*. Once the database schema is created, endpoints for login, logout, and registration are established. Users must have the *admin* role to access the database modification capabilities; however, they can modify the data they have imported.

4.7 Results and Use Cases

This section presents an overview of the results we have obtained and explains the use cases that have been implemented within the platform. These use cases demonstrate the platform's ability to operate across various fields and highlight the advantages of digital twins in urban environments.

4.7.1 City2Twin Platform

The City2Twin platform is designed to be simple and intuitive, enabling users to navigate efficiently while offering enhanced visualization features, as shown in **Figure 4.18**.

To ensure the platform's capability in handling large datasets, we imported a CityJSON LoD 2 file of Zurich city from the client side, containing approximately 200,000 objects and sized at 300 MB. The model was successfully loaded. The page size reached up to 1.7 GB.

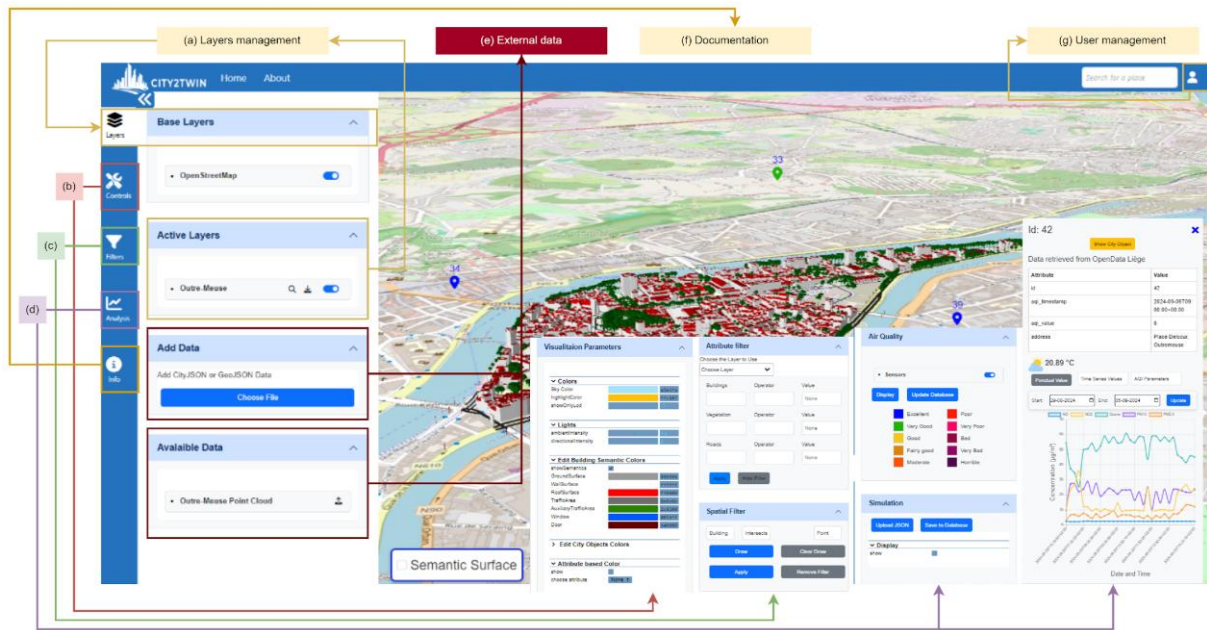


Figure 4.18. The City2Twin overview: (a) layer management panel, (b) controls panel, (c) attribute and spatial filters, (d) urban analysis (air quality and energy simulation panel), (e) import external data, (f) documentation, (g) user management.

4.7.2 Use Cases

Under the analysis module, two applications are presented (see **Figure 4.19**). The first section contains the application dedicated to air quality analysis, while the second focuses on energy simulation in buildings and that is further developed in chapter 5.



Figure 4.19. Air quality measurement stations classified based on the AQI value.

Air Quality Analysis

The dynamic data are used to perform air quality analysis. As explained in the methodology, data is retrieved via an external API and can either be stored or directly visualized. This near real-time visualization informs users of the current air quality in the area. Regarding time series data from the database, various analyses can be conducted.

The X-axis can represent dates and times in chronological order, different hours of the day, or days of the week, while the Y-axis can represent temperature, AQI values, or concentrations of pollutants. The data can also be filtered for a specific period. By combining different parameters on the X and Y axes and adjusting date ranges, a more comprehensive and advanced air quality analysis can be performed (see Figure 4.20). For example, it is possible to identify the times of day or days of the week when air quality is at its best, or simply to track the evolution of a parameter over time. Seasonal trends can also be evaluated if sufficient data is available in the database. These analyses support decision-makers and governments in making informed choices to improve citizens' quality of life.



Figure 4.20. Time series analysis of air quality: (a) Concentrations of each pollutant at time t; (b) Temperature variations from 08/01 to 08/26; (c) Average hourly concentrations calculated between 08/19 – 08/26; (d) Concentration variations between 08/23 and 08/25; (e) Average daily concentrations calculated between 08/19 – 08/26; (f) Average hourly temperatures calculated between 08/20 – 08/25.

Energy Simulation

Simulation results are typically provided in CSV or JSON files. These files contain key-value pairs, where the key represents the building ID, and the value is the associated result. Although a real simulation cannot be performed due to the lack of necessary data in our 3D model, an approach has been developed to support data from these simulations. For this purpose, we created a JSON file containing data for testing (considered as a result of a simulated energy demand). This file can be injected into the model for visualization, as shown in **Figure 4.21**, and can also be stored in the database as an attribute of urban objects.

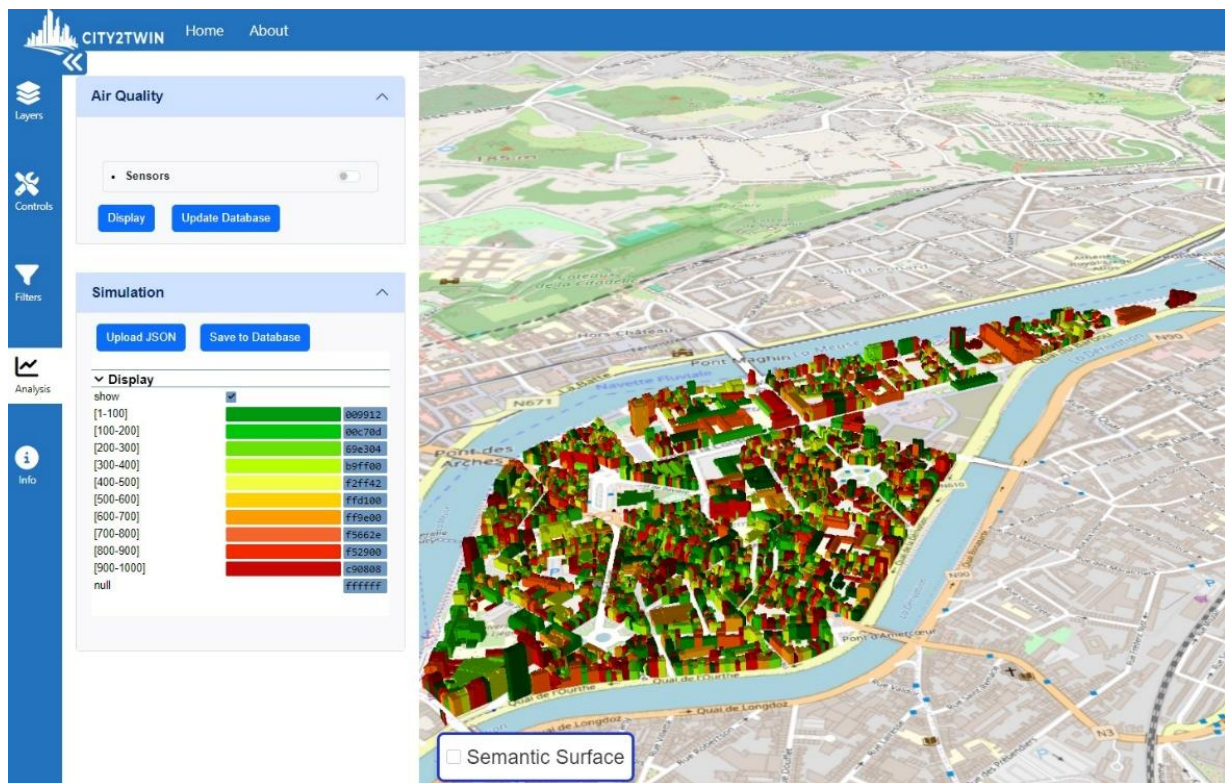


Figure 4.21. Visualization of integrated simulation data on the model.

4.8 Discussion

This section analyzes the results of the UDT implementation, focusing on both the effectiveness and the limitations of the developed tool.

CJDB is designed to improve query efficiency by keeping the data model as simple as possible. The advantages of CJDB are its simple table structure, small storage size and easy to work with: easy data import, simple query design and no need for multiple tables joins to get the target result.

4.8.1 Used Tools

Based on the tests conducted on both databases, we observed the importance of having a simple table structure, like the one proposed by CJDB. However, one of the challenges encountered with this database is the issue of schema consistency. Unlike 3DCityDB, where attributes are stored in a separate table with constraints on variable types, in CJDB,

they are stored in a JSONB column in the same table as the urban objects. Consequently, no constraints on the attribute value types are applied. We had to implement these constraints manually; for instance, when updating an attribute value, it first compares the type of the old value with that of the new one. Furthermore, importing data to CJDB needs further optimization, namely updating the *ground_geometry* for certain objects that have empty multipolygons. Moreover, the CJDB export tool exports data to a file, which must then be reloaded into the python code, which is not optimal as it increases the server response time.

Although CesiumJS is the most widely used library, Three.js was chosen for this project to reduce loading times and improve performance while maintaining a direct link between the client and the database. Additionally, Three.js enables compatibility with CityJSON, allowing us to benefit from its various advantages, such as support for handling semantic surfaces, which is very challenging to achieve with CesiumJS and the 3D GeoJSON format.

4.8.2 Evaluation compared to Cesium-based approach

To assess the contributions of this project, we compared **City2Twin**, following our methodology, with a CesiumJS-based approach (see **Table 4.7**). The comparison was conducted under the same configuration for both approaches: CJDB as the database, a Flask-based server, and Google Chrome as the client browser.

Table 4.7. Comparison between City2Twin and CesiumJS based approach.

	CesiumJS based approach	Giro3D based approach
Approach	Direct	
Database	CJDB	
Supported Data	3D GeoJSON	CityJSON, CityJSONL
Loading time	Slower	Faster
Memory	Large (> 3 GB)	Light (< 1 Gb)
Geospatial Features	Advanced	Advanced
Attribute Filters and Manipulation	Not Supported	Supported
Semantic Surface Manipulation	Not Supported	Supported
Data Import and Export from Client Interface	Not Supported	Supported
Integration of Multiple data and different CRS	Not Supported	Supported
Visualization controls and rendering styles	Less Advanced	Advanced

After testing the two workflows, the Cesium-based approach retrieves 3D objects from the database as 3D GeoJSON via an SQL query and visualizes them in a Cesium scene. Database retrieval is fast; however, loading and rendering the resulting GeoJSON in Cesium is slow. For the ThreeJS-based approach, CJDB exports 3D objects in CityJSON Lines format. Because ThreeJS does not natively support CityJSON Lines, the exported data must first be converted to CityJSON using the *cjio* or *CjSeq* libraries before it can be visualized. The database export time is negligible, particularly with *CjSeq*, which is approximately ten times faster and loading the resulting CityJSON file in a ThreeJS scene is also fast. An overview of the workflow is illustrated in **Figure 4.22**.

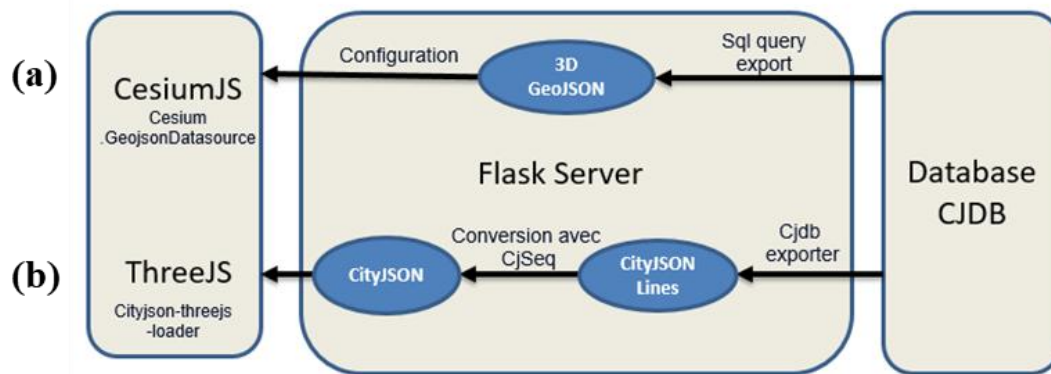


Figure 4.22. Overview of the visualization architectures of the two approaches: (a) the CesiumJS-based approach and (b) the Giro3D framework-based approach.

Based on this comparison, we conclude that City2Twin has introduced significant improvements in data analysis, particularly with attribute and spatial queries, as well as the semantic surfaces. Regarding attribute manipulation, it offers a more intuitive interface compared to the way attributes are modified in Ninja and also implements constraints on value types to reinforce data consistency.

The developed approach has demonstrated the importance of having a 3D database that directly interacts with the client interface, allowing for the manipulation or modification of urban objects without relying on manual exports and file hosting on a server (indirect approach). In addition to this database-based approach, it also supports files imported from the client interface (such as 3D point clouds, CityJSON, or GeoJSON). Thus, it can function as a simple 3D visualization tool on the web without the need for database configuration.

Although dynamic data can be injected into the model itself via an extension, the use of the STA Database has facilitated the management and storage of this data. This standard not only allows for the storage of data but also includes the management of various elements related to *datastreams* with their descriptions, such as the sensor used, the observed properties, the station and its location, etc.

The developed framework is a generic framework that can be used for other applications requiring dynamic data or not: e.g., mobility, energy demand, urban planning, etc. All the tools and solutions used in this project are open-source and free, facilitating the development of extensions for specific needs.

4.8.3 Evaluation of City2Twin flexibility and consistency

This section evaluates how the City2Twin design choices prioritize flexibility for evolving UDT workflows, and what this implies for semantic consistency and validation.

CityJSON as a lightweight data model

City2Twin adopts CityJSON as its core data model. This choice was motivated by the objective of exploring whether CityJSON can serve as a suitable reference data model for lightweight and web-oriented UDT implementations.

CityJSON provides a structured representation of city objects and geometry while remaining compact, readable, and directly usable in web environments. Unlike XML-based encodings, it can be parsed, visualized, and manipulated easily in client applications without complex conversion pipelines. This makes it particularly suitable for interactive platforms where users are expected to inspect and modify urban data.

In addition, CityJSON supports extensions and custom attributes, which is essential for UDTs where requirements evolve over time and new properties must be introduced.

CJDB instead of 3DCityDB

For data storage, City2Twin uses CityJSON Database (CJDB), which relies on PostgreSQL and JSONB to store CityJSON features. This choice differs from traditional CityGML-based database solutions (i.e., 3DCityDB). While the design of 3DCityDB provides strong schema-level consistency, it also increases complexity, maintenance effort, and the cost of evolving the data model. CJDB proposes a simple and lean alternative. It relies on a small number of tables and stores attributes and geometries directly in JSONB columns. Geometries follow the CityJSON structure and adopt a Simple Feature paradigm. This significantly simplifies the database schema while retaining access to both semantic and geometric information.

The choice of CJDB in City2Twin was motivated by the need for flexibility, incremental updates, and compatibility with web-based workflows.

Benefits of CJDB for UDT implementation

The use of CJDB in City2Twin brings several practical benefits:

Flexibility: attributes remain in JSON format regardless of their name or structure, allowing new properties to be added without schema migrations.

Reduced schema complexity: the database structure is simpler and easier to maintain compared to 3DCityDB.

Web compatibility: CityJSON objects can be retrieved and consumed directly by client applications.

Incremental updates: objects and attributes can be modified individually, which supports lifecycle-oriented UDT workflows.

Performance for interactive use: CJDB supports spatial indexing and JSONB indexing, enabling efficient spatial filtering and attribute access for web applications.

These properties make CJDB well suited for evolving UDT implementations, such as City2Twin.

Trade-offs of the current implementation (flexibility versus consistency)

The City2Twin implementation highlights some limitations. While CJDB is relational in the sense that it relies on PostgreSQL, it deliberately minimizes schema-level constraints. As a result, the database does not enforce strong semantic consistency rules.

Attribute naming conventions, data types, unit consistency, and cross-object invariants are not guaranteed by the database schema itself. Consequently, consistency cannot be treated as an inherent property of the storage layer in City2Twin. Instead, it becomes an operational concern that must be explicitly addressed elsewhere in the system.

In City2Twin, consistency is managed primarily at the server-side application level, implemented through the Flask backend. The backend acts as a control layer between the database and client applications, ensuring that all updates pass through a controlled interface. In practice, this includes type handling and validation before writing attributes to JSONB; checks on object existence before updates; controlled update logic for semantic structures to avoid unintended global modifications; explicit handling of update operations rather than unrestricted database access. Client-side validation mechanism complements this approach by preventing user errors and improving usability. However, the client is not relied upon as the authoritative mechanism for consistency, as client logic can be bypassed.

This architecture illustrates a key principle: in flexible UDT designs, consistency must be actively enforced by the application, rather than assumed to be guaranteed by the database schema.

Validation as prerequisite

City2Twin assumes that the CityJSON datasets are valid before they are integrated into the system. Since CJDB stores geometry and attributes as JSONB, the database does not enforce CityJSON schema validity or geometric correctness. Validation of the city model therefore belongs to the ingestion workflow rather than the storage layer itself. Making this explicit is important for operational use, because invalid input data would otherwise propagate to the client and affect visualization and update operations.

Robustness

Beyond semantic consistency, flexibility also impacts robustness. In City2Twin, updates are performed through controlled API endpoints, but complex operations highlight the importance of transaction handling and error management. Without careful control, partial failures could leave the system in an inconsistent state. This further reinforces the role of the application layer as a boundary in flexible UDT architecture.

Indexing and performance considerations

Although CJDB provides indexing mechanisms, these were not used in the current City2Twin deployment. The CJDB design, by storing attributes and geometries in JSONB, makes indexing flexible but also different from strict relational indexing (i.e., indexes must be chosen based on expected query patterns rather than enforced by the schema).

In City2Twin, since indexes were not deployed, the observed performance reflects a default configuration rather than an optimized setup. This highlights that robustness and responsiveness in web-based UDTs depend not only on the data model and storage solution, but also on concrete deployment choices such as which indexes to create and maintain.

Versioning, provenance, and long-term maintenance

The remaining limitation of the current implementation concerns versioning and provenance tracking. While City2Twin supports incremental updates, attribute changes overwrite previous values unless explicitly preserved. For long-term UDT operation, explicit versioning mechanisms at the application or database level would be required to track the evolution of urban data over time.

Overall, City2Twin shows that CityJSON and CJDB form a solid basis for lightweight UDT implementations, provided that validation, consistency, and robustness are treated as explicit design concerns rather than implicit guarantees of the storage layer.

4.9 Conclusion

This project focused on developing a framework for implementing a UDT, covering key aspects such as data integration, storage, visualization, and analysis. It has led to a functioning open UDT, **City2Twin**, that tackles the gap in literature regarding technical requirements and enhances UDT development while taking advantage of lightweight standards and solutions. The developed framework is based on the CityJSON encoding. This format offers several advantages due to its lightweight nature and simplicity, facilitating its manipulation and integration into a visualization platform. For data storage and management, after various tests and comparisons between CJDB and 3DCityDB, the CJDB database was chosen for its simplicity, lightweight nature, and compatibility with an approach based on CityJSON encoding. Direct interaction with the database is a major advantage, allowing for continuous maintenance and efficient data updates without the need to export files manually. Additionally, the use of a separate database (STA Database) with the CityThings concept prevents the direct storage of dynamic data within the model, which can already be heavy. It also enables effective management of the various elements related to *datastreams*.

In conclusion, the City2Twin framework demonstrates a robust approach to urban management through the integration of 3D city models and dynamic data. Although this work successfully met our main objective, future improvements could focus on optimizing data retrieval methods and expanding the platform's capabilities to support more advanced urban analysis and several types of data.

CHAPTER 5: Levels of Integration within UDTs: Implementations and Evaluation

Preface

As Urban Digital Twins increasingly move towards large-scale and federated ecosystems, supported by European initiatives and funding instruments, data integration and interoperability remain among the key technical barriers. While many UDT implementations are still carried out at pilot, district or city-scale, the long-term vision toward interconnected and federated digital twins raises fundamental questions regarding how heterogeneous data sources can be integrated, aligned and maintained across different lifecycle stages and governance settings. Tackling these questions requires grounding the conceptual framework developed earlier in chapter 3 and examining how integration is implemented in practice.

One of the main reasons for persistent integration and interoperability challenges is the absence of an official, dedicated standard for UDTs. As a result, current UDT implementations rely on various standards, tools and design choices, mainly driven by data availability, domain specific constraints or requirements. Despite this diversity, current implementations are still built upon semantic 3D city models, which historically embed semantic information through Application Domain Extensions mechanisms in CityGML-based workflows. From a conceptual perspective, such approaches correspond to level 1 of integration, where data are integrated by extending the underlying data model. However, experiences have shown that the ADE- based approach introduces significant complexity in practice, including increased schema management overhead, limited tools support. While ADEs provide a powerful mechanism at the model level, their practical implementations have often proven difficult to scale and reuse across UDT workflows.

Within the UDT development framework, there is a growing need for lightweight, web-based integration approaches. These approaches support modular architectures and distributed components, particularly when UDTs evolve over time. In this context, OGC web standards provide practical building blocks for interoperability through service-based architecture. The OGC API family offers a promising foundation to tackle data integration challenges while providing modern, RESTful interfaces that enable consistent access and exchange of geospatial data across systems.

Building on the conceptual framework of data integration levels, and on the City2Twin implementation, this chapter focuses on understanding how these integration levels are implemented in practice and how they shape UDT workflows and outputs. The aim of this chapter is to analyze how different integration levels are implemented in practice through concrete technical choices and to what extent they support operational UDT requirements across the UDT lifecycle. This chapter therefore aims to clarify why data integration matters in UDTs, how integration levels shape architectural decisions, and how they affect the robustness, maintainability, and interoperability of UDT systems.

Motivated by the technical heterogeneity, this chapter investigates what each integration level enables in terms of use cases, as well as the trade-offs it introduces. Particular attention is given to the implications of integration choices when moving toward more distributed and federated UDT architectures, where data remains under the control of their providers and are accessed through standardized interfaces rather than centralized

pipelines. In such context, integration is no longer only a technical issue, but also involves non-technical aspects such as governance rules, access rights, licensing, and data sharing terms, which can change over time.

To ground this analysis in practice, the chapter implements the proposed Levels of Integration within the City2Twin platform. Integration is illustrated and detailed through a set of representative scenarios, each corresponding to a specific data source and use case commonly encountered in UDT implementations. A scenario is defined here as a concrete integration approach applied to a given type of data at a specific integration level and lifecycle stage, designed to support a particular analytical or operational objective. Three scenarios are considered: (1) integration of BIM model (i.e., IFC models), (2) integration of dynamic IoT sensor data, and (3) integration of simulation outputs. These scenarios are not intended to be exhaustive, but they reflect widely adopted practices and allow a systematic exploration of how integration levels are applied in real-world UDT workflows.

This chapter aims to investigate the following research questions: [RQ5. How can different levels of integration be implemented across various UDT use cases?](#) And [RQ6. How can a flexible and lightweight UDT architecture support, illustrate, and operationalize these data integration levels?](#)

The contributions of this chapter are twofold. First, it operationalizes the Levels of Integration framework by implementing it through concrete, reproducible workflows within an Urban Digital Twin platform. Second, it derives technical insights and practical guidelines that can support researchers and practitioners in making informed integration choices when designing UDTs. By linking conceptual integration levels with their technical implementation, the chapter aims to contribute to a more operational understanding of data integration in Urban Digital Twins and to support the transition toward interoperable and federated UDT ecosystems.

Based on the article (Jeddoub et al., 2026, under review)

Why Integration Levels Matter in Urban Digital Twins: A Comprehensive Study of Multisource Data Integration.

Jeddoub, I., Hajji, R., & Billen, R. Why Integration Levels Matter in Urban Digital Twins: A Comprehensive Study of Multisource Data Integration. *International Journal of Applied Earth Observation and Geoinformation* ([Under Review](#)).

Abstract: Multisource data integration has long been central to urban and geospatial communities, with standards developed to foster interoperability, reuse and automation. Yet, in Urban Digital Twins (UDTs) which currently rely on adapting existing standards rather than developing a specific UDT standards, the integration approaches and their technical challenges remain under-explored. Since UDTs operate over a lifecycle (creation, use, and update) that incorporates heterogeneous datasets, it is fundamental to understand how data are integrated in practice and which mechanisms best support different lifecycle stages. We introduce a Levels of Integration (LoI) framework to conceptualize and implement UDT data integration. Level 1 extends data model via extensions or profiles; Level 2 focuses on harmonizing and adapting datasets to conform to a schema model without altering the model itself; and Level 3 links data through front-end architecture and standard-based APIs. The framework is application-agnostic, designed to cover diverse datasets and tools, and is demonstrated through reproducible workflows from integration to visualization and analysis. In contrast to prior work, which rarely analyzes in depth the technical requirements of integration level, this article provides a comprehensive reference that clarifies why integration levels matter in UDTs. Furthermore, beyond using existing tools, the chapter introduces detailed workflows and provides a technical reference for multi-source integration across the UDT lifecycle, formalizing practical guidelines for researchers and practitioners, including attention to identifiers and semantics, validation beyond schemas, and updating mechanisms that reflect the evolving nature of UDT systems. The proposed LoI framework offers a reproducible basis for current and future UDT deployments.

Keywords: Urban Digital Twins; Data integration; CityGML; CityJSON; IoT; Simulation.

5.1 Introduction

Urban Digital Twins (UDTs) have gained significant attention as transformative tools for understanding, managing, and optimizing complex urban systems through the coupling of physical and digital environments. UDTs originate from the convergence of geospatial technologies, Building Information Modelling (BIM), City Information Modeling (CIM), Internet of Things (IoT) infrastructures, and advanced analytics and provide dynamic, data-rich representations of cities that support real-time decision-making, simulation, and long-term planning (Deng et al., 2021). Their potential spans multiple domains from infrastructure management and mobility optimization to environmental monitoring and participatory governance (P. Alva et al., 2022).

However, the implementation of UDTs also introduces significant technical and semantic challenges, namely regarding data integration and interoperability. Urban data are inherently heterogeneous, deriving from various sources remote sensing, cadastral databases, IoT sensor networks, BIM models, Geographic Information System (GIS) and crowdsourced information (Jeddoub et al., 2023). Integrating such disparate datasets within a unified digital framework requires not only technical alignment of data schemas, temporal resolutions, and semantics, but also institutional coordination and governance across data providers and governance structures (Billen et al., 2014; Noardo, 2022). Achieving interoperability across systems and stakeholders thus remains one of the most critical hindrances in realizing operational and up-to-date UDTs (Lei et al., 2023). Similar integration challenges have long been recognized and tackled to bridge GIS and BIM domains to enable coherent 3D city modelling and information sharing (Liu et al., 2025). Yet, with the emergence of UDTs and their relevance for many cities and territories worldwide, the scope and complexity of integration have expanded considerably. UDTs need to keep the continuous synchronization of static and dynamic data throughout their lifecycle from creation and data ingestion, through operation and analysis, to update and maintenance (Ketzler et al., 2020). Consequently, understanding how data integration occurs across these lifecycle stages, and at what levels, starts to be fundamental to ensure that UDTs are both scalable and sustainable.

Beyond architectural and technical solutions, emerging data governance paradigms, namely data spaces, federated data platforms, and open data ecosystems are increasingly recognized as foundational components of next-generation UDTs (Kogut and Heijden, 2025). Urban Data Spaces, such as the European Data Strategy²⁷ and GAIA-X²⁸, offer distributed yet interoperable infrastructures that foster sharing, integration and reuse of data from multiple domains and stakeholders under common governance frameworks. Similarly, federated and open data platforms enable cities to maintain local control and data sovereignty while preserving semantic interoperability and cross-domain collaboration (Gil et al., 2024; Herzog et al., 2025). These infrastructures not only facilitate data integration at scale but also promote transparency, inclusivity, and innovation by enabling new services and analytics built upon shared urban data ecosystems.

²⁷ <https://digital-strategy.ec.europa.eu/en/policies/strategy-data>

²⁸ <https://gaia-x.eu/>

The increasing focus on standards is highlighted by (Song and Le Gall, 2023) which explicitly identify interoperability, model versioning, and data-exchange protocols as persistent limitations to digital twin applications. In the urban and geospatial domain, standardization efforts such as IFC, OGC CityGML, OGC API standards and FIWARE continue to play a central role in aligning data models and exchange protocols across domains. The interplay between these standards and emerging federated data infrastructures highlights a shift toward purpose-driven and governance-aware integration, where the level and method of integration are determined not only by technical needs but also by how the data will be used, who controls it, and under what rules it is shared. Earlier groundwork in OGC Testbed-18 (2022)²⁹ demonstrated standard-based web architecture for domain-specific urban workflows particularly building-energy analytics and visualization. It showed how OGC API standards can be connected to make heterogeneous 2D, 3D, sensor datasets and computations accessible and interoperable across platforms. Building on this foundation, recent OGC work Urban Digital Twins Interoperability Pilot (UDTIP, 2025)³⁰ applies an API-centric integration architecture to live smart-city scenarios, fusing persistent 3D GeoBIM with dynamic sensors and trajectory streams via the OGC API family.

These findings present a roadmap for both the evolution of standards and the design of adaptive, interoperable UDT architecture. Furthermore, these OGC initiatives demonstrate a progressive evolution of conceptual and technical frameworks for interoperability across multiple levels from single-UDT implementations to federated, service-oriented integration that enables city-scale, cross-domain digital ecosystems and supports large-scale implementations across scales.

In parallel, recent studies indicate that Generative AI (GenAI) and Large Language Models (LLM) based interfaces can directly support data integration and semantic interoperability in UDTs. (Xu et al., 2024) highlights how GenAI can autonomously generate and complete urban datasets, synthesize scenario and design variants, and translate natural-language queries into structured operations, thus introducing cognitive capabilities into existing integration workflows. Furthermore, (Kanna and Kolbe, 2025) demonstrate the use of LLMs to automatically extract and align attributes from heterogeneous data sources and integrate them into CityGML 3.0 models through an enrichment pipeline. Their approach enhances the completeness and semantic consistency of urban datasets and reduces manual harmonization efforts, while directly strengthening interoperability. These studies reveal a progressive shift in UDT integration from standard-driven and API-centric architectures toward AI-enabled, semantically and human-inclusive ecosystems capable of sustaining interoperability across scales and lifecycle stages.

Building on this evolving context, data integration is a significant technical challenge that hinders the full implementation of UDTs. Given their data-driven nature, UDTs rely on the integration of heterogeneous data sources, schemas, and structures. This process is

²⁹ <http://www.opengis.net/doc/PER/T18-D012>

³⁰ <http://www.opengis.net/doc/PER/UDTIP>

inherently complex due to the diversity of techniques and systems involved. While integration challenges have traditionally been explored in similar urban and geospatial domains, few studies have investigated in detail the specific aspects of data integration in UDTs. In addition, the proliferation of UDT initiatives shows that the focus has shifted from acquiring new data to exploring how existing data can be integrated, restructured, and connected to meet the evolving requirements of UDTs.

Motivated by the heterogeneity of technical workflows and drawing on the findings of (Jeddoub et al., 2024), our aim is to analyze this concept of integration levels and its relevance to UDTs and to raise fundamental questions about [what each integration level offers to the use cases, their respective pros and cons, and how it shapes outputs and workflows](#). Addressing these questions clarifies how integration levels are conceptualized and implemented in real-world use cases, which will support wider UDT adoption in practice. This paper aims to operationalize the concept of data integration levels within UDTs by linking conceptual findings with their practical implementation across representative data sources (i.e., BIM models, IoT streams, and simulation outputs). This work seeks to demonstrate how these levels can be implemented in practice through concrete integration workflows. To this end, we define three integration scenarios, each corresponding to a distinct use case and different data sources.

Our contributions are twofold: (1) highlighting the relevance of the integration levels of UDT and operationalizing them by defining concrete workflows and use cases; and (2) proposing comprehensive, reproducible workflows that illustrate each level and derive practical guidelines for practitioners embarking on UDT implementations.

This chapter is structured as follows. Section 5.2 provides a review of existing literature on data integration in UDTs, outlines current frameworks, identifies challenges, and integration levels discussed in previous studies. Section 5.3 defines and explains the integration scenarios and their associated use cases. Section 5.4 presents the implementation and conducted tests, including the technical details, integration workflows, and results, followed by an analysis of the outputs. Section 5.5 discusses the findings, summarizes practical insights, and suggests recommendations for future implementation. Finally, Section 5.6 concludes the work, highlights the main contributions and suggests directions for future research toward holistic and comprehensive data integration frameworks within UDTs.

5.2 Background and Related work

5.2.1 Data Integration Overview

Data integration is the core operational layer of UDT implementations. It supports their ability to represent, synchronize, and simulate complex urban systems. It relies on the harmonization of heterogeneous datasets (i.e., unstructured, semi-structured, and structured) originating from multiple domains. While early research explored interoperability within BIM-GIS integration and IoT-GIS convergence, these discussions are increasingly reframed within the paradigm of UDTs, where integration is no longer an optional step but a fundamental and continuous process across the UDT lifecycle (Piras et al., 2024; Xia et al., 2022).

Recent conceptual studies have formalized the levels of interoperability and maturity that characterize UDT systems. Drawing from cyber-physical system theory, Acharya et al. (2024) identify interoperability as a multi-layered property spanning technical, syntactic, semantic, pragmatic, organizational dimensions, each associated with persistent challenges such as schema heterogeneity, temporal synchronization, and data provenance. Moreover, (Lu et al., 2020) underline that integration in the built environment must harmonize diverse data modalities within one unified and coherent digital ecosystem. In these frameworks, interoperability is conceptualized as a necessary layer prior to any digital twin implementation, enabling seamless interaction between physical entities and their digital representations, regardless of the underlying architecture or domain.

Furthermore, Haraguchi et al. (2024) has also examined UDTs under the umbrella of maturity levels. They introduce a maturity model for assessing the governance models of digital twins' cities, evaluating how governance structures, data strategies, and technical infrastructures evolve across eight stages. Similarly, the analysis between digital models, digital shadows, and digital twins based on the degree of coupling between the physical and digital counterparts has become a foundational framework for understanding digital twin evolution (Fuller et al., 2020; Kritzinger et al., 2018). These classifications emphasize the increasing complexity of data exchange, synchronization, and feedback mechanisms as systems evolve from static representations to fully bi-directional models. However, while maturity and coupling levels have been widely discussed, there remains limited attention to the levels of data integration that mainly help to address how heterogeneous datasets are technically, semantically, and temporally combined within UDTs, and how such integration unfolds across their lifecycle stages.

To address integration more concretely, several standardized and architectural approaches have been advanced. Within the geospatial community, CityGML 3.0 has emerged as a semantic hub that integrates heterogeneous urban models (BIM, GIS, point clouds), supporting unified 3D street-space representation and improved cross-domain reuse (Deng et al., 2025). Complementary efforts, such as the Urban Model Platform (Herzog et al., 2025) advance a public, standards-based backbone for modeling and simulation in UDTs. This approach enables the decentralized and federated integration of diverse models, together with explicit model-to-model communication, and thus moves beyond monolithic twin architectures toward modular, multi-model ecosystems.

In parallel, research on data governance has highlighted the organizational and infrastructural dimensions of integration. (Yan et al., 2025) propose a Common Data Environment (CDE) architecture that structures data flow around three key enablers: sources, management components, and consumers. Within this framework, five data integration approaches were analyzed (i.e., schema, ontology, service, system and process and tool-based methods) based on representative examples. (Channi et al., 2025; Gil et al., 2024; Kogut and Heijden, 2025) further highlight that sustaining interoperability relies on transparent models that maintain data quality, provenance, and consistency while supporting distributed collaboration through data spaces and federated infrastructures. Despite these advances, integration practices remain fragmented, and use-case driven. Most UDT implementations are guided by data availability or project requirements rather than predefined, lifecycle driven integration mechanisms (Lu et al., 2020). Persistent challenges include semantic inconsistencies between models, lack of temporal

synchronization for dynamic data, limited scalability of interoperability mechanisms, and less documented methodological frameworks for cross-domain twin federation (Lei et al., 2023). Consequently, although existing research extensively addresses interoperability, maturity, and coupling, limited attention was given to how data are integrated in practice, at what levels, and through which mechanisms across the UDT lifecycle. Addressing these questions is essential to move from conceptual frameworks to operational, reproducible, and sustainable urban digital ecosystems.

5.2.2 Data integration levels

Based on the analysis of the UDT implementations, three main data integration levels are defined. Level 1- Conceptual Data Model (CDM): extends, profiles, or merge the existing conceptual model to incorporate new information and features required by specific applications (e.g., UDT use cases). Level 2- Database Level (DB) aligns datasets with the existing conceptual model and standards without changing the underlying schema. This ensures new data conforms to predefined structures while preserving the semantic consistency of the urban information model. Level 3- Front-End Integration integrates data through various system architectures and interfaces such as GIS tools, web applications, geoportals, dashboards, linked data, federated platforms, data spaces, and game engines. This level enables visualization and management of 2D and 3D data directly in modern web browsers, addressing challenges of standardization and syntactic interoperability. Beyond visualization, it supports customized applications for urban planning, infrastructure management, and environmental monitoring by harmonizing diverse data sources within a connected platform. Jeddoub et al. (2024) propose a conceptual framework that defines data integration levels mainly identified in literature and in practice, structured according to the UDT lifecycle. **Figure 5.1** illustrates these integration levels across the lifecycle. Each lifecycle phase is associated with a data layer that represents its inputs. In the creation phase, building the Digital geoTwin (modeling and descriptive twin) primarily requires static geodata and dynamic data. Once the geoTwin is created, the use phase links it to specific operational use cases and enriches it with external sources or by coupling it with domain-specific twins and simulation models (e.g., energy, mobility, environment). These couplings generate derived datasets that are fed back into the UDT for analytics and decision-making. The update phase is particular. The level of integration is related to which component to be updated (i.e., the geo-base twin or the domain specific twin). In practice, the effective integration level in this phase is guided by the levels previously deployed during creation and use. This framework is illustrated earlier through existing UDT initiatives taking into consideration the heterogeneity of datasets, lifecycle stages, and the diverse use cases each UDT serves.

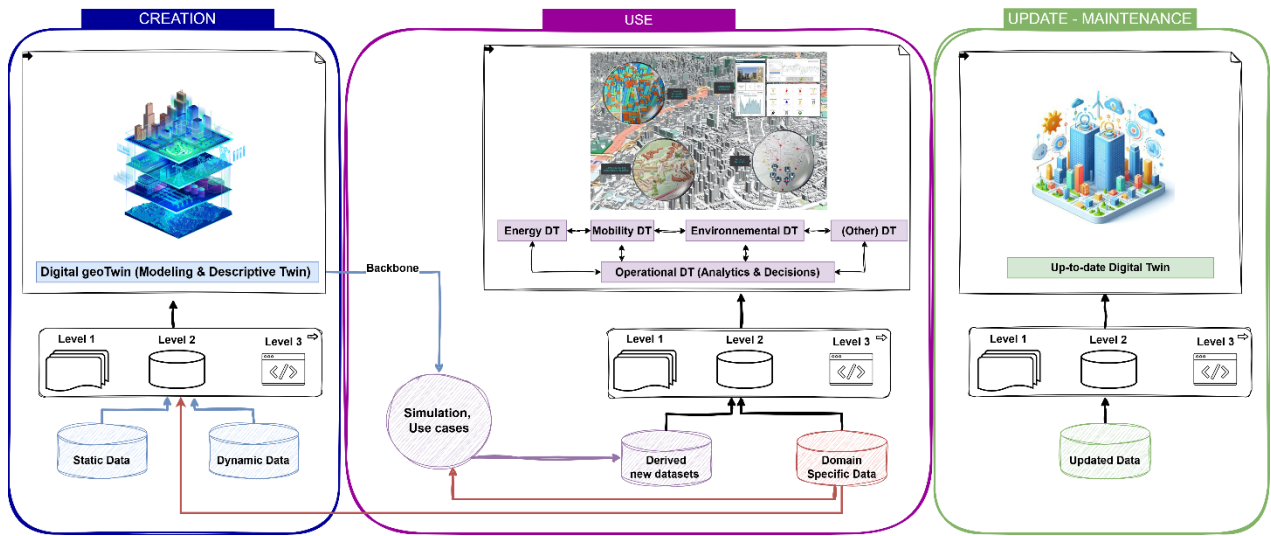


Figure 5.1. Unpacking the integration levels across the UDT lifecycle.

5.3 Conceptualization and scenario definition

This section aims to provide conceptual context for the integration levels. Although scientific literature considers various integration approaches, it lacks a multidimensional analysis of these integration levels within UDTs. We therefore conceptualize integration through scenarios definition. A scenario is defined as an integration approach of specific data at a given level during a lifecycle stage, designed to meet a specific use case. Accordingly, in this section, we define scenarios based on the data source and the corresponding uses cases, we then detail our data integration approaches for each scenario. The scenarios proposed are the most adopted in practice, yet they have not all been implemented and validated across the proposed integration levels. In the following section, we will focus on three of the most common data types in the context of UDTs to better illustrate our concern. The selection of data is not arbitrary; it reflects both data availability and the use case we implement. Accordingly, three scenarios are defined: (1) Scenario 1: Integration of IFC model, (2) Scenario 2: Integration of IoT data into UDT, (3) Scenario 3: Integration of simulation outputs into UDT. In general, these integration levels are intended to be applicable to any type of data and use case.

5.3.1 Scenario 1: Integrate an IFC model

Integrating BIM and GIS for UDTs is widely recognized as a critical step toward fostering interoperability and supporting standardization. Over the past decade, GeoBIM experts have explored a variety of integration techniques that aim to bridge these two paradigms. Integration techniques are generally categorized into enrichment, conversion, combination (fusion), and linked data (Noardo et al., 2021). Despite significant progress, several challenges remain unresolved, including semantic heterogeneity, discrepancies in spatial resolution and scale, and inconsistencies in data formats. In response, recent research increasingly advocates moving beyond the conventional discourse of unidirectional BIM-to-GIS conversion toward more systematic, application-driven workflows that address these issues in a more comprehensive way.

A key insight emerging from this line of research is that integration should be guided by the purpose it serves understanding why BIM–GIS integration is performed, and for what application, allows the integration mechanism to be selected in a way that preserves the strengths of both domains.

For instance, in **Figure 5.2**, the same IFC model is integrated differently across the UDT lifecycle, depending on the purpose. During creation (planning and visualization), IFC is mainly integrated following client-side approaches. The client or platform layer can view and interact with both native IFC and 3D city models. In use or operation phase, semantic alignment (Level 1- CDM, e.g., IFC ADE) supports analysis that needs rich building semantics such as indoor navigation. During update phase, converting the IFC model to CityGML/ CityJSON would be relevant at city scale and the changes flow back to the UDT via ETL process, keeping the city model synchronized. The integration level is thus phase and use-case-driven, preserving the strengths of BIM and city models throughout the UDT lifecycle. **Figure 5.2** provides a concrete example of how BIM data can be integrated into the UDTs. It is not the only possible integration pathway; many scenarios exist but those are the most adopted in practice when implementing UDTs.

Accordingly, our scenarios definition is intended as an illustrative configuration, demonstrating the applicability of integration levels of BIM within UDTs.

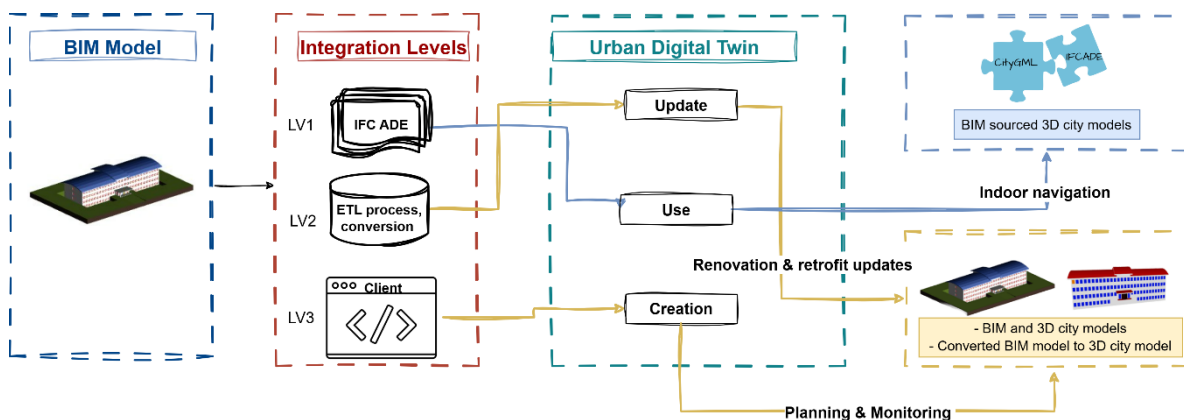


Figure 5.2. Examples of BIM model integration levels across UDT phases.

In our case, integrating IFC models into a UDT platform serves the specific purpose of updating and enriching 3D city models. This raises question about which level is sufficient to retain semantic and geometric richness without unnecessary complexity. By tackling this question, this study aims to illustrate the integration mechanisms that balance the complementary advantages of BIM-GIS and align them with the requirements of UDT applications.

To illustrate our concern about updating the UDT, we adopted three integration approaches, each following different technical workflow (refer to section 5.4.1).

5.3.2 Scenario 2: Integrate IoT Data into UDT

Integrating IoT data provides dynamic components to UDTs, upgrading them from static models into continuously synchronized models of the physical world. Recent advancements in sensing and data collection, via IoT sensors and wearable devices,

enable cities to capture and stream high-frequency measurements into UDTs for downstream urban analysis. Furthermore, IoT and sensor data are often coupled with predictive models or used to validate simulation outputs for UDT applications, while supporting continuous monitoring of urban features and infrastructure. Many cities are now equipped with networked sensors and devices on roads, buildings, and vehicles that collect real-time data. These streams enable monitoring and what-if scenarios for mobility, environment, water, energy, and public space. However, realizing this potential depends on interoperable standards, robust data governance, and reliable edge-cloud pipelines to address data heterogeneity, latency, coverage gaps, and security concerns.

Despite progress in standardization, semantic discrepancies, spatial and temporal resolution mismatches, and inconsistent formats still hinder effective integration; current efforts to bring IoT into 3D city models and UDTs therefore focus on the application first. Geospatial and web standards help address interoperability, using the OGC SensorThings API to publish and consume observations on the web. With an application first focus (choosing sensors and semantics to meet the use case), cities can integrate IoT streams into UDTs seamlessly, enabling end-to-end workflows from sensing to modelling to operational actions and evidence-based decisions.

For our use case of air quality analysis, many scholars have converged on a standardized pipeline that integrates heterogeneous sensor feeds into 3D city models. (Gitahi and Kolbe, 2024) propose a standardized framework that integrates dynamic data via the OGC SensorThings API (STA), links it to semantic 3D city models through CityGML (including the CityGML 3.0 “Dynamizer” concept for time dependent properties) and delivers 4D web visualizations in a CesiumJS viewer, enabling users to explore and compare scenarios (e.g., air quality alongside mobility). At the city scale, the OGC 3D-IoT pilot demonstrates the same principles across indoor and outdoor models: $PM_{2.5}$ and room occupancy observations are exposed via STA; features are served through OGC API - Features using IndoorGML and 3D Tiles or glTF; clients include dashboards and AR viewers; and server-side processes (OGC API - Processes) interpolate observations to derive fields such as air quality above a street block.

Figure 5.3 showcases how IoT streams are integrated across the UDT lifecycle. During creation (planning and visualization), data are incorporated at runtime in the client to produce quick view and inspection without altering the core model. In the use/operation phase, semantic alignment via the “Dynamizer” ADE extends the data model so sensor measurements become time-varying attributes of city objects well suited to today’s context where many 3D city models are still CityGML 2.0 thereby enabling alerts and analytics. For updates/synchronization and evaluation, CityGML 3.0 Dynamizer (file or 3DCityDB with ETL) integrates time-series data linked to stable feature IDs, supporting before and after assessment and historical analysis. As illustrated in **Figure 5.3**, IoT data can be integrated at different levels which is consolidated from current case studies in practice. This represents and summarizes the common possibilities to integrate IoT data into UDT rather than an exhaustive framework that covers the whole possibilities (i.e., integration can be performed at a single level or across multiple levels). Thus, the integration level is guided by the UDT stage and is use-case-driven, preserving both IoT stream fidelity and city-model coherence. Building on this analysis, we illustrate our air quality use cases across three integration levels (refer to Section 5.4.2).

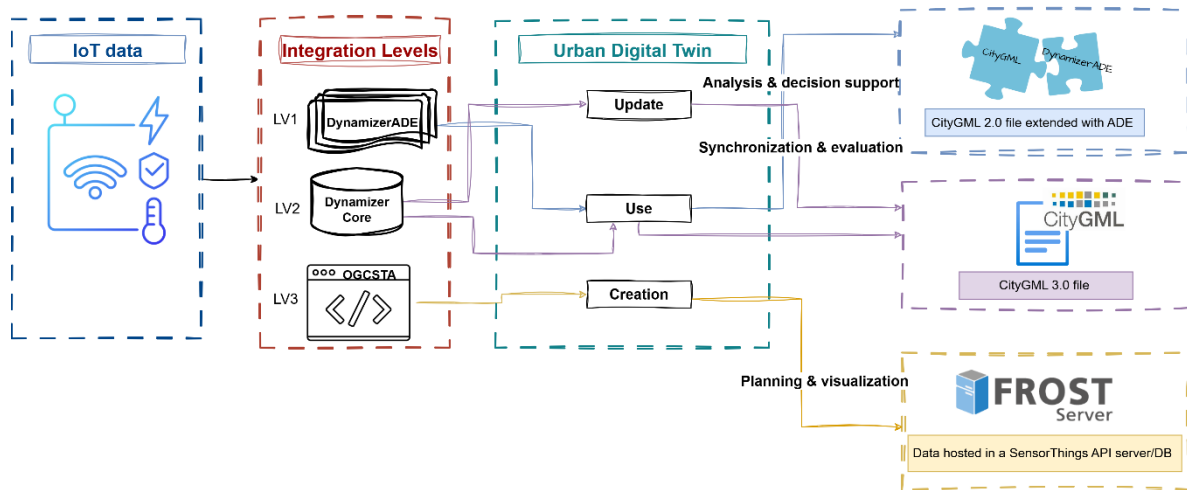


Figure 5.3. Illustrative approaches of IoT feeds integration levels across UDT lifecycle.

5.3.3 Scenario 3: Integrate simulation outputs into UDT

Simulation outputs (e.g., heating, cooling demands, solar potential, greenhouse-gas emissions) are among the most common analytical datasets integrated into UDTs. UDTs provide a unified and geospatial data infrastructure to analyze, simulate, and visualize building-scale energy demand and emissions across various spatial and temporal scales, thereby informing policy design, retrofitting strategies, and monitoring (Miralles-Wilhelm, 2016; Saad & Eicker, 2023). Depending on the use case and the modeling approaches, simulation outputs may take diverse forms ranging from time-series profiles to complex Computational Fluid Dynamics simulations that capture pollutant dispersion, heat transfer, and wind, to name a few.

The integration of simulation outputs into UDT can be achieved across three levels and relies mainly on the use case and could configure two or more integration levels. The level 1- CDM of integration involves the use of ADEs, such as the Energy ADE or the Food-Water-Energy (FWE) ADE (Agugiaro and Padsala, 2025; Padsala et al., 2021). The aim is to store data using standardized data models ensuring the semantic representation of building energy properties, system configurations, and simulation outputs. This level enables interoperability and reuse across systems and applications, which is relevant to the synergy between Building Energy Simulation (BES) and UDTs. Recent work presents a UDT framework for evaluating Positive Energy District (PED) potential built on OGC standards and the newly developed CityGML Energy ADE 2.0 (Padsala et al., 2025). The Energy ADE v2.0 enhances the old model by introducing a “Resources module” for representing multi-sectoral energy flows and an “Urban Function Area” module for aggregating data across spatial units such as neighborhoods or districts, enabling comprehensive multi-scale energy analysis (Agugiaro and Padsala, 2025).

Level 2- DB of integration shifts the focus toward adapting simulation outputs, so they meet UDT requirements. A common practice is the use of CityGML Generics to store application-specific attributes when no suitable thematic class exists. While this offers flexibility in data representation and management, it heavily relies on ad hoc generic attributes, which lead to duplication of effort and hinder interoperability across tools. Another approach at Level 2 is to integrate the datasets into a 3DCityDB already extended with the ADE. The energy-related results are written directly into the ADE tables that are

already registered in the database. This lets simulation tools feed and update the database from their outputs without modifying the source CityGML files, and it avoids many pitfalls of purely file-based updates and exchanges. This method improves data retrieval, consistency, and downstream visualization.

Level 3- front-end is achieved through dynamic data exchange via open standards and APIs, including the OGC SensorThings API (STA), Web Feature Service (WFS), and broader Spatial Data Infrastructures (SDIs). This architecture allows real-time synchronization between physical observations, simulations, and UDT visualizations, supporting adaptive and interactive analyses. Building on this, recent work leverages OGC APIs to implement fully interoperable SDIs for UDTs, using modern RESTful endpoints that streamline discovery, access, and processing across distributed urban data sources. These standardized interfaces enable on-the-fly processing pipelines, seamless connections to modeling tools, and efficient delivery to web clients' capabilities that are essential at this integration level (Thunyathep Santhanavanich et al., 2022). Furthermore, recent studies propose lightweight, modular, standardized architectures that enable efficient handling, storage, and visualization of enriched urban energy datasets within energy-UDT frameworks (Jeddoub et al., 2025).

In **Figure 5.4**, simulation results are integrated according to each phase and purpose. In the creation phase, Level 3- front-end integrates scenarios through a customized viewer or OGC SensorThings API services and retrieves them at runtime, enabling quick visualization and analysis. In the use/operation phase, there are two options: Level 2- DB (CityGML Generics) when only lightweight simulation outputs (e.g., annual heat demand, or PV yield) are needed, attached as simple attributes; and Level 1- CDM (Energy ADE) when richer semantics are required for deep analysis, such as PED, GHG ,system archetypes, schedules, zone results, or scenario variants stored in an Energy ADE-extended file/DB. For update, the outputs are disseminated through Levels 2 and/or 3- ETL processes write updated datasets to, for instance, a CityGML generics file, or to the client/services layer, which integrates them back for further analysis. In **Figure 5.4**, we highlight common integration scenarios derived from literature and current implementations. These are examples, among other possible scenarios, for integrating simulation data into UDTs.

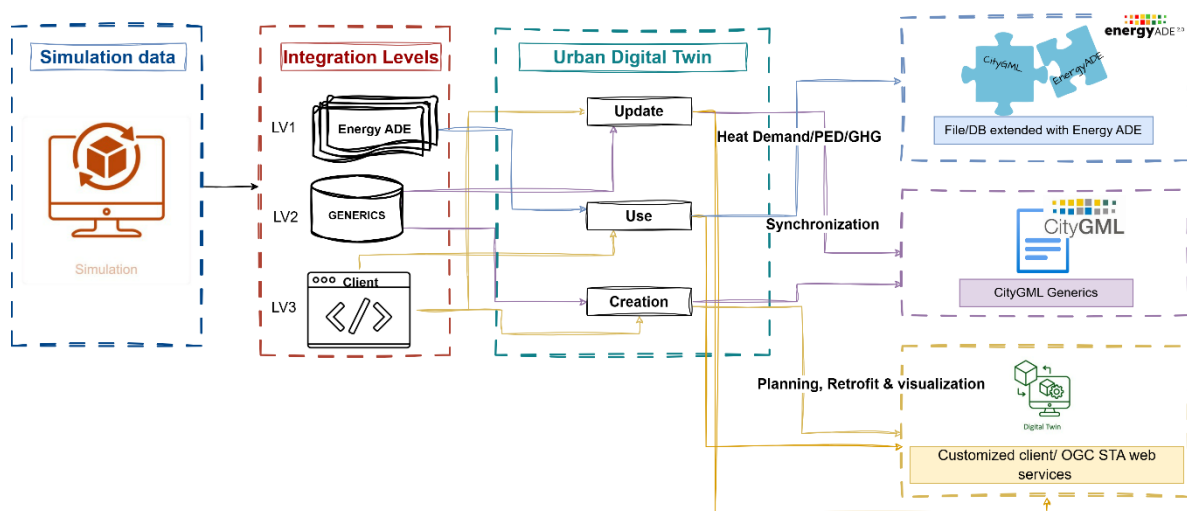


Figure 5.4. Examples of integrating energy-simulation outputs across the UDT lifecycle.

In this article, we focus on a single use case: building-specific heating demand. We will implement all the three integration levels accordingly (refer to Section 5.4.3).

5.4 Use cases Implementation

In this section, we detail the technical approach used to implement the scenarios. We first describe the datasets and key preprocessing steps, then outline the software tools and standards employed, and finally present the end-to-end workflow from data processing to integration and visualization. We then report and comment on the results for each scenario and integration level, followed by a discussion of advantages and limitations. Results and discussion are presented primarily through figures and tables, respectively. For testing purposes, most of our implementations are conducted within City2Twin platform developed in-house. It serves as our test environment, representing an open-source framework specifically developed for UDT applications supporting various data formats.

5.4.1 Scenario 1: Integrate IFC model to UDT

Used Datasets

For testing purposes, we selected two publicly available Building Information Models: AC20-FZK-Haus, representing a small single-family house, and AC20-Institute-Var-2, (see **Figure 5.5**) depicting a multi-storey institutional building (refer to Table 5.1). These open-source IFC datasets are well-established in the research community and are frequently used to benchmark interoperability and data conversion workflows. Their availability and completeness made them ideal candidates for testing our implementation.

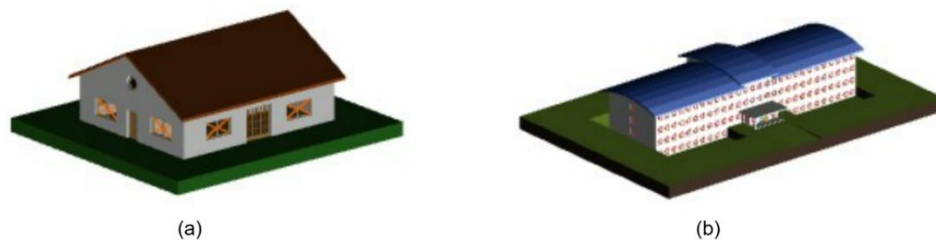


Figure 5.5. The IFC models, (a) AC20-FZK-Haus and (b) AC20-Institute-Var-2.

Table 5.1. BIM model overview.

IFC Model name	IFC version	Object count	Building type	Source
AC20-FZK-Haus	IFC4	102	Single house	KIT
AC20-Institute-Var-2	IFC4	896	Office building	KIT

Software used and workflow

For Level 1- CDM, we intended to demonstrate this data integration level by running the IFC2CityGML converter tool. However, the tool is proprietary, and we were given access only to the IfcADE XSD, which supports schema-driven visualization (e.g., in FME Data Inspector) but is insufficient to execute a conversion (Biljecki et al., 2021; Stouffs et al.,

2018). The IFC2CityGML³¹ workflow requires a local BIMserver, a conversion client (GUI), and JSON graph-rule mappings, none of which were available to us (refer to the user guide³²). Consequently, we could not perform the implementation. To discuss feasibility and expected behavior at this level of integration, we grounded our analysis in prior results from the GeoBIM project (Task 4), which documented how IFC-to-CityGML 3.0 conversions behave in practice (including geometric/semantic validation limitations and loss of some elements) (Francesca Noardo et al., 2020). In that benchmark, the IFC2CityGML converter was applied to IFC4 architectural datasets targeting LOD3/4 CityGML outputs, the same scale and characteristics we intended to test (building-level IFC models with typical representation types and attributes), so the benchmark serves as a suitable proxy for our context (Francesca Noardo et al., 2020).

Given this close alignment of dataset scope and model characteristics, using the GeoBIM findings as evidence for expected outputs and limitations in our setting is relevant to our intended work. Our discussion therefore synthesizes the benchmark's reported outcomes of converting IFC to CityGML 3.0 and outlines how, had the converter been accessible, we would have executed and validated the pipeline in our environment.

For level 2- DB, we use the ifc-to-citygml 3.0 FME workspace. The FME workflow is designed to convert IFC datasets into CityGML 3.0 while defining mapping rules from IFC to their corresponding CityGML 3.0 classes. We run this workspace on both benchmark datasets, AC20-FZK-Haus and AC20-Institute-Var-2. Our tests illustrate how an ETL approach performs for this use case; however, the generated CityGML file is not valid due to geometric issues, indicating inconsistent shell orientations that render the solids invalid and hinder reuse in external applications.

To ensure compatibility with our visualization platform which imports CityJSON files, we used the IfcEnvelopeExtractor tool. This open-source software enables the automatic conversion of IFC models into CityJSON v2.0 files across multiple Levels of Detail (LoDs). In this study, we used the last release, version 0.2.6, which supports several IFC schema versions (i.e., IFC2x3, IFC4, and IFC4x3).

To configure the conversion process, we used the Python-based GUI that allows users to define parameters, manage input and output files, and monitor the conversion workflow. The GUI interacts directly with the IfcEnvelopeExtractor library, simplifying the process and ensuring reproducibility. Before conversion, both models were georeferenced using the IfcGRef toolkit³³. This step is to ensure spatial alignment and consistency with geospatial coordinate reference systems. The georeferencing process is based on the processes suggested in previous works, which emphasize the integration of surveyed control points and accurate placement of IFC models within real-world coordinate systems (Hakim et al., 2024; van der Vaart et al., 2025, p. 2).

By establishing georeferencing, we ensured that the resulting CityJSON files are correctly placed. Subsequently, the georeferenced IFC models were processed using the IfcEnvelopeExtractor. The configuration follows the recommendations outlined in the

³¹ <https://ifc2citygml.github.io/>

³² <https://ifc2citygml.github.io/files/UserGuide.docx>

³³ <https://ifcgref.bk.tudelft.nl/templates/guid.html>

tool's documentation and related publications (van der Vaart et al., 2024, 2025). The output was CityJSON files generated at different LoDs, which were then integrated into our visualization platform for further analysis and evaluation. Although the same workflow is tested on both BIM models, our discussion focuses primarily on the AC20-Institute-Var-2 model. This model produced valid and consistent results at LoD 3.2, which is particularly relevant to our targeted use case involving detailed building envelope representations and urban-scale visualization.

For level 3- front-end, we implemented the approach following two main steps designed to handle, analyze, and visualize IFC models. First, we extended the City2Twin platform capabilities to handle IFC files, enabling their import, parsing and interactive visualization within a web environment using OpenBIM Components and Giro3D frameworks, thereby supporting efficient rendering, structured exploration, and interaction with model entities. Second, the system retrieves and integrates semantic attributes via a Python backend built on Flask. Upon upload, the backend performs semantic extraction at the element level collecting identifiers (e.g., GlobalId), descriptive attributes (e.g., Name, PredefinedType), and quantitative measures from property sets and returns a structured JSON response. The frontend then merges these semantics with the visual layer, linking each geometric component to its associated metadata. This workflow (i.e., geometric visualization followed by semantic enrichment) provides a scalable, web-based approach that preserves IFC semantics while enabling interactive analysis.

Results and Discussion

Based on our tests, the integration levels produce different outputs (refer to **Figure 5.6**). We summarize the key advantages and limitations of each level in **Table C.1**, explaining where each approach fits best, with a specific focus on semantic consistency and updating mechanisms for the UDT.

For Level 1- CDM, although we did not test the conversion ourselves, we can still draw conclusions about data integration at Level 1 according to the GeoBIM findings: (1) IFC to CityGML 3.0 conversions remain feasible but are constrained by available rule sets and validation tools; (2) geometry and semantic validation limits and some missing elements are reported; and (3) workflow prerequisites (BIMserver, conversion client, graph-rule mappings) are critical requirements of running the tool. Based on that, we were able to derive some insights about how we would configure, test, and validate the pipeline once the converter becomes accessible. The tool can produce CityGML 3.0, but with limited software support for CityGML 3.0, outputs were assessed primarily through visual inspection and partial geometry checks, rather than full semantic or geometric validation.

For Level 2- DB, we focus on the outputs of the IfcBuildingExtractor applied to the IFC model AC20-Institute-Var-2. After the conversion, the CityJSON model was checked and validated. For our update use case, we keep only the LoD 3.2 version to support the update use case. We then imported the converted CityJSON into the City2Twin platform. In the viewer, users can switch from the existing LoD 2 to the updated LoD 3.2. Throughout the conversion, IFC attributes for each semantic element were preserved (e.g., walls, roofs, slabs, and openings retained their original BIM properties while being harmonized with the CityGML and CityJSON data model). Overall, this reflects the aim of Level 2, adapting IFC data to match the existing schema and re-injecting updates into the UDT without

altering the conceptual model. The conversion process demonstrated the strength of level 2 for maintaining semantic and geometric consistency between the two paradigms. The LoD3.2 abstraction aligns well with the CityGML/CityJSON refined LoDs. The approach also allows for continuous updating when new or renovated IFCs become available, they can be processed through the same pipeline and seamlessly integrated into the UDT.

For level 3- front-end, using the extended City2Twin platform, we successfully loaded the IFC model AC20-Institute-Var-2 in the web browser, rendered it interactively, and linked each IFC element to its semantic data coming from the Flask backend. Users can select, inspect key attributes (e.g., GlobalId, Name, PredefinedType) and parsed property sets / quantities, with a clear mapping from the 3D representation to its metadata. The viewer supports search by class, isolate/hide, and section views for inspecting interiors. The backend returns a structured JSON per element, and the frontend merges this with the geometry layer. Models are imported reliably; georeferencing is correct (where present in IFC), and units are handled consistently. This level delivers a web-based approach where IFC models are visualized and analyzed directly without converting it first, while keeping semantic granularity. This level extends the UDT by allowing direct BIM interaction in the same environment used for city-scale visualization. It enables detailed inspection of building components while maintaining full semantic fidelity, bridging the gap between the converted CityJSON model (Level 2) and the native BIM source. Users benefit from a consistent interface where both the LoD 3.2 city model and the original IFC are accessible, ensuring connection between urban and building scales.

Based on our tests and for Level 1- CDM, on the GeoBIM findings, the three integration levels yield distinct outputs, each with specific advantages and limitations. Level 1 (IFC to CityGML via ADE) enables transfer of a selected subset of IFC semantics into CityGML to support domain use cases without importing full BIM complexity; however, outcomes depend on rule-set coverage and IFC representation types, tools for CityGML 3.0/ADEs remains limited, validation is largely visual or partial, and reproducibility depends on operational prerequisites.

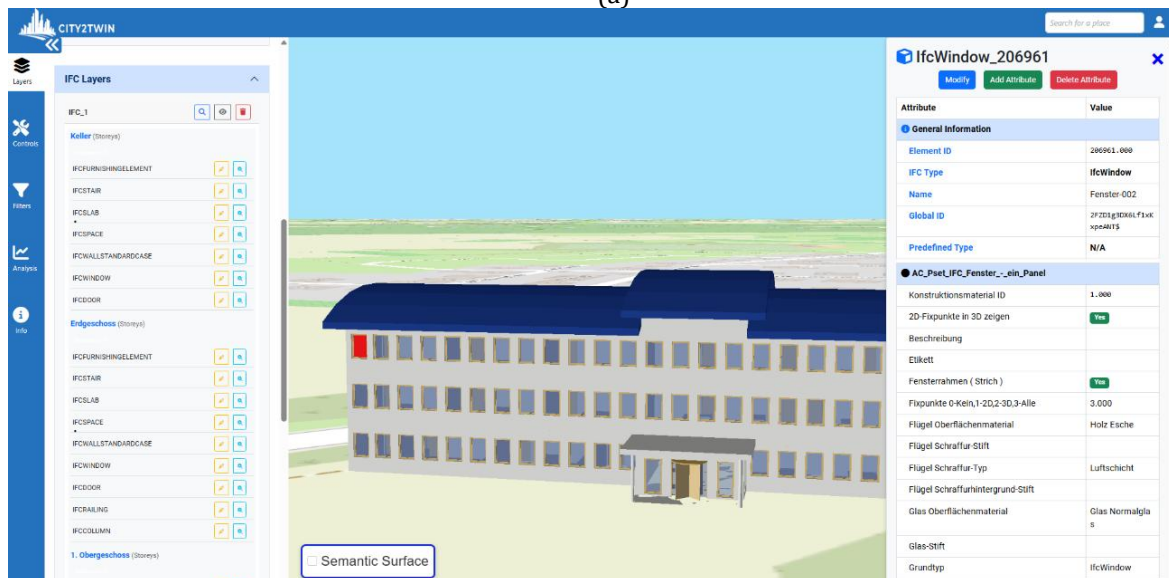
Level 2- DB (IFC to CityJSON harmonization) produces lightweight, schema-compatible city models that retain key BIM semantics, integrate cleanly into City2Twin, and support regular updates to the UDT; the limitations are potential loss of semantic granularity where no direct CityJSON equivalent exists, reliance on accurate preprocessing (i.e., georeferencing and model quality), and the need for re-conversion when updates occur.

Level 3- front-end (native IFC in the UDT) preserves full BIM semantics for interactive web-based querying and analysis and scales to collaborative use, but it demands higher rendering and memory resources for complex models, longer parsing times with deep property sets, dependence on correct IfcMapConversion for georeferencing, and it offers reduced harmonization for urban-scale analytics compared with CityJSON.

These levels offer complementary pathways: Level 2 is most effective for regular UDT updates and city-scale visualization, Level 3 for detailed inspection and analysis of BIM content, and Level 1 when formal ADE-based exchange is needed despite current tools and validation constraints.



(a)



(b)

Figure 5.6. Scenario 1 outputs: (a) Level 2-LoD 3.2 CityJSON model, (b) Level 3 -AC20-Institute-Var-2 IFC model.

5.4.2 Scenario 2: Integrate IoT data into UDT

Used Datasets

The dynamic dataset used in this study consists of near real-time air-quality measurements for the city of Liège provided by the ISSeP API. Fixed monitoring stations mounted on lamp posts record timestamped concentrations of pollutants, along with temperature (°C) and the sensors position/metadata. Measurements are ingested and stored every 5 minutes, enabling fine-grained spatial and temporal analysis. The variables, units, temporal granularity, access methods and brief descriptions are summarized in Table 5.2.

Table 5.2. Summary of dynamic air-quality datasets.

Data	Source/Provenance	Observations	Units	Temporal granularity	Access/API
Sensors position and Air quality APIs	ISSeP (Public Service Scientific Institute)	Date and time of the measurements and concentrations of PM _{2.5} , PM ₁₀ , O ₃ , NO ₂ , NO; Temperature	µg/m ³ (pollutants), °C (temperature)	As provided by ISSeP API (timestamped measurements) for storing every 5 minutes	ISSeP APIs: measurements (latest values) and sensors positions

Software used and workflow

In this scenario, we implemented our methodology using the same dataset, where lamp posts are represented as GenericCityObjects within a CityJSON 2.0 model. To achieve level 1-CDM of integration, we designed a CityJSON Extension³⁴ inspired by the CityGML 3.0 Dynamizer concept into a lightweight, web-oriented format. The extension was upgraded and validated for CityJSON 2.0 following the official CityJSON guide³⁵ ensuring schema compliance and compatibility with modern CityJSON viewers. Structurally, the +Dynamizer can be defined either as a single object or an array, allowing multiple pollutant or environmental variables, such as PM_{2.5}, PM₁₀, NO, NO₂, O₃, and temperature, to coexist as independent temporal components within the same CityObject. Each Dynamizer entry comprises its unit of measure, an optional linkToSensor (referencing the original SensorThings source), and minimal metadata. On the server side, a Flask service connects to the SensorThings (PostgreSQL) database, it maps each Datastream_ID or Multi_Datastream_ID to the corresponding CityJSON attribute, retrieves values and writes the latest observations into the static attributes while maintaining the full time series under +Dynamizer.

On the client side, we extended the visualization environment to support both static and dynamic representations of the same CityObject. The user can toggle between the lamppost's geometric and semantic attributes and its dynamic components. The IoT manager layer handled by AttributeManager and IoTDataHandler, detects +Dynamizer entries and renders bar charts for punctual values and line charts for time series, including optional hourly or weekday aggregations. This provides an intuitive interface for exploring the spatiotemporal analysis of environmental measurements directly within the UDT.

At Level2- DB, we converted the lamppost CityJSON to CityGML 3.0 with citygml-tools³⁶, validated the output, and represented the object as CityFurniture. We then added CityGML

³⁴ <https://github.com/JEDDOUB/Extensions/blob/main/dynamizer-v2-structure.ext.json>

³⁵ <https://www.cityjson.org/tutorials/upgrade20/>

³⁶ <https://github.com/citygml4j/citygml-tools>

Generics attributes for PM_{2.5}, PM₁₀, NO, NO₂, O₃, and temperature, giving each variable a stable attribute that can be referenced by Dynamizers through attributeRef while preserving interoperability across tools. To integrate IoT data from our OGC STA service, we used two schema-compliant patterns: (1) for variables currently exposed via MultiDatastreams, we embedded observations as GenericTimeseries, resulting in valid files without external response mappings; (2) for temperature, which is available as a single Datastream, we attached a live SensorConnection that references the Datastream ID and the STA base URL, enabling runtime updates and validation. We adopted this approach since the validator supports SensorConnection for single Datastreams but not the ExternalService response-mapping approach required for MultiDatastreams. So, we embed data for MultiDatastream variables and keep live connections only for attributes that have their own Datastream.

At Level 3- front-end, the integration moves beyond static data enrichment to establish a bidirectional link between the 3D city model and real-time environmental data streams. This level adopts the CityThings concept, which links OGC-STA entities with the CityJSON 3D City model, enabling a dynamic UDT that continuously synchronizes sensor observations and urban features. In this configuration, each sensor is modeled as a Thing in the STA data model, holding a unique identifier (city_object_id) corresponding to the CityObject (e.g., lamp post) stored in the CityJSON DB (CJDB) static database. The STA database provides a standardized and interoperable structure for managing real-time observations. Instead of injecting dynamic data directly into the CityJSON file, all observations are stored in a separate STA PostgreSQL database, ensuring scalability and efficient updates without overloading the 3D model. A flask middleware serves as the bridge between these two databases. It interfaces with the STA database using SQL queries to retrieve and store air-quality measurements. Data is fetched typically every five minutes and inserted into the observations table following the STA schema. This real-time integration mechanism ensures continuous synchronization between the physical sensors and their virtual counterparts, thereby enabling automatic UDT updates. On the client side, the City2Twin platform retrieves the latest sensor observations directly through REST API. Each sensor, represented as a GenericCityObject within the CityJSON model, is visualized in the UDT and dynamically linked to its corresponding STA entity. The platform dynamically requests data from the STA service, parses JSON responses, and visualizes time series via Charts, offering multiple temporal analysis by timestamp, hourly average, or daily. This seamless integration of CityJSON, CJDB, and STA under the CityThings architecture establishes a fully operational Level 3 integration for UDT, fostering real-time integration, standardized data management, and interactive visualization in a unified open-source framework.

Results and Discussion

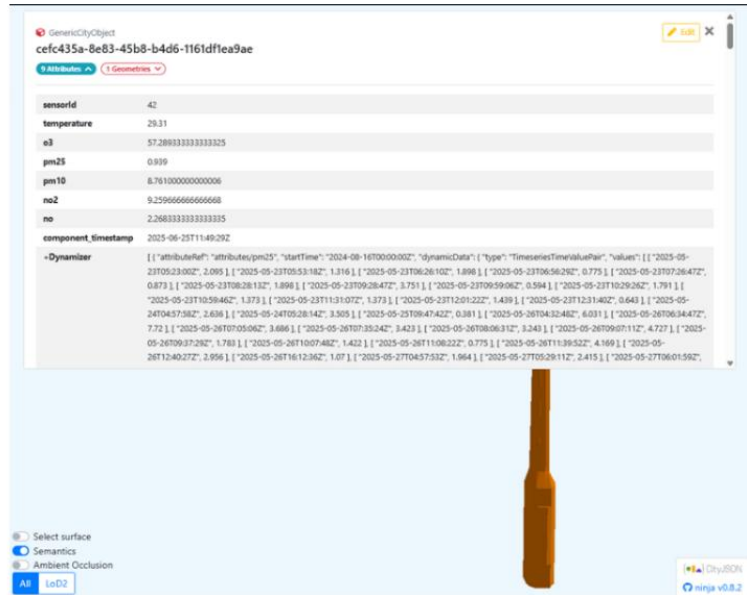
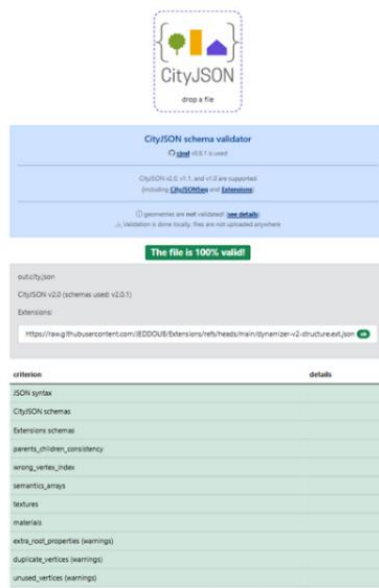
The integration approaches were implemented and evaluated using the same dataset and 3D model of lamp posts. At Level 1- CDM, we created a CityJSON 2.0 file extended with the “Dynamizer” mechanism, enabling each lamp post to store both its static attributes and dynamic air quality measurements within a single file (refer to **Figure 5.7(a, b)**). This extension follows the conceptual principles of the CityGML 3.0 Dynamizer semantics and provides a compact, unified representation of dynamic data. As summarized in **Table C.2**, this approach offers several advantages, mainly simplicity and full semantic consistency, but it also presents some limitations, particularly in scalability and real-time

synchronization. Since time series are directly embedded in the CityJSON file, time series quickly increase the file size, and each update requires full file rewriting. Furthermore, while the “Dynamizer” provides more flexibility in representing temporal information, it remains relatively restrictive in terms of data access and interoperability support.

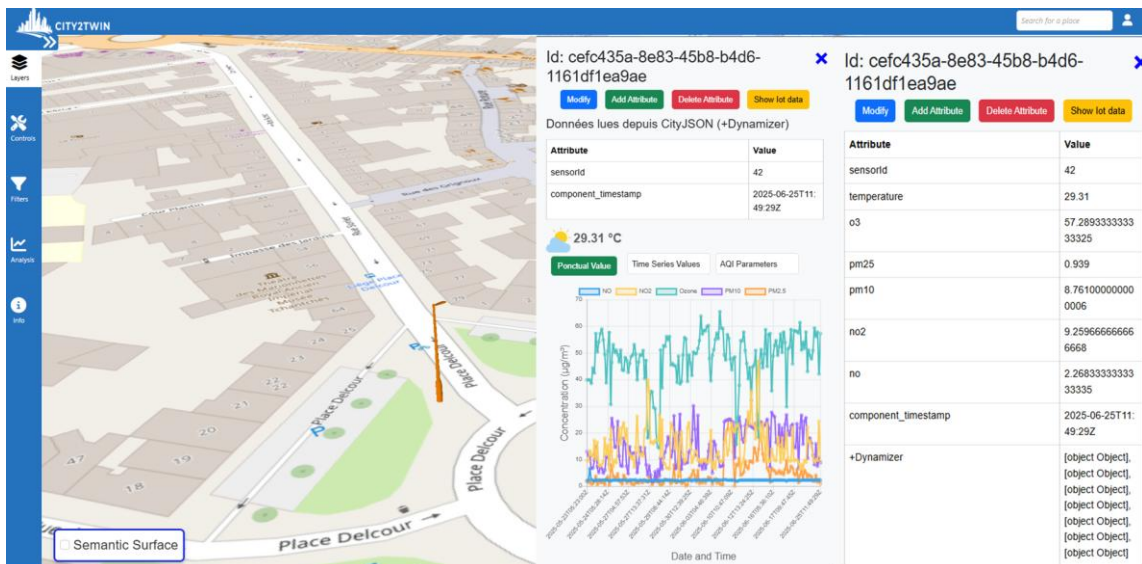
For level 2- DB, the resulting CityGML 3.0 files can be imported into 3DCityDB v5 for further analysis or visualized using the 3DCityDB-Web-Map-Client. Static generics attributes are queryable after import, whereas Dynamizer attributes (embedded time series and live connections) require tools that support such data to resolve and display the temporal values at runtime. For exploratory visualization, we used FME Data Inspector to inspect the CityGML attributes and confirm the presence and structure of the “Dynamizer” elements (refer to **Figure 5.7c**). In short, Level 2 delivers a standards-compliant, database-ready CityGML representation. The model can be persisted and viewed, and the dynamic attributes can be visualized once a viewer or service implements the CityGML 3.0 Dynamizer logic for fetching (SensorConnection) or reading (GenericTimeseries) the time-varying values.

At Level 3- front-end, the integration was extended to a real-time system architecture using the CityThings concept (Santhanavanich and Coors, 2021), where each lamp post is linked to its corresponding SensorThings API (STA) entity stored in a PostgreSQL database (see **Figure 5.7d**). The link between the CityJSON model and the STA schema was implemented through Flask middleware that manages data ingestion, synchronization, and updates of air-quality measurements. City2Twin dynamically requests the latest observations through REST endpoints and displays them in real-time charts. This configuration provides significant advantages in scalability, interoperability, and analytical capability, as it leverages a standardized OGC-compliant structure for sensor data management. However, as also outlined in **Table C.2**, it requires a more complex infrastructure, with dependencies on network connectivity, STA database, and middleware stability. In contrast to the “Dynamizer”, the FROST-Server implementation of STA provides a simpler but highly standardized structure for representing observations.

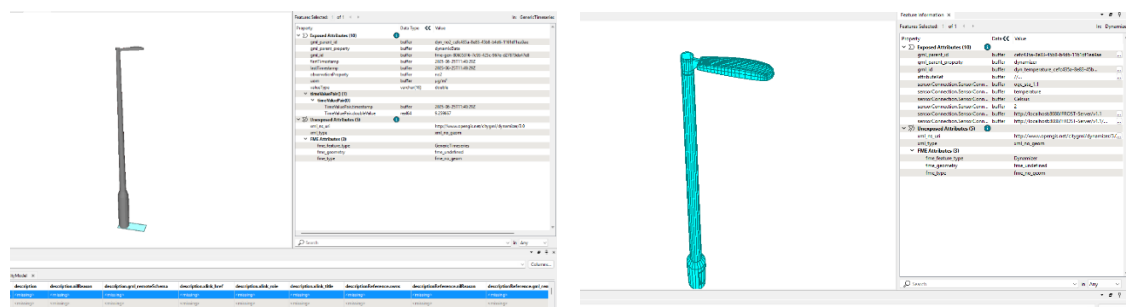
Overall, the comparison highlights three complementary approaches. Level 1- CDM proposes a lightweight, semantic CityJSON model that is suitable for standalone or offline use. Level 2- DB upgrades the representation to CityGML 3.0, keeping the lampposts as CityFurniture with Generics and attaching dynamics via “Dynamizer” embedded GenericTimeseries for MultiDatastream variables and a live SensorConnection for temperature producing valid and database-ready files. Level 3- front-end supports an operational, scalable, and continuously synchronized UDT. These integration approaches illustrate a progressive transition from file-based approach to service-oriented data integration paradigms within the CityJSON framework.



(a)



(b)



(c)

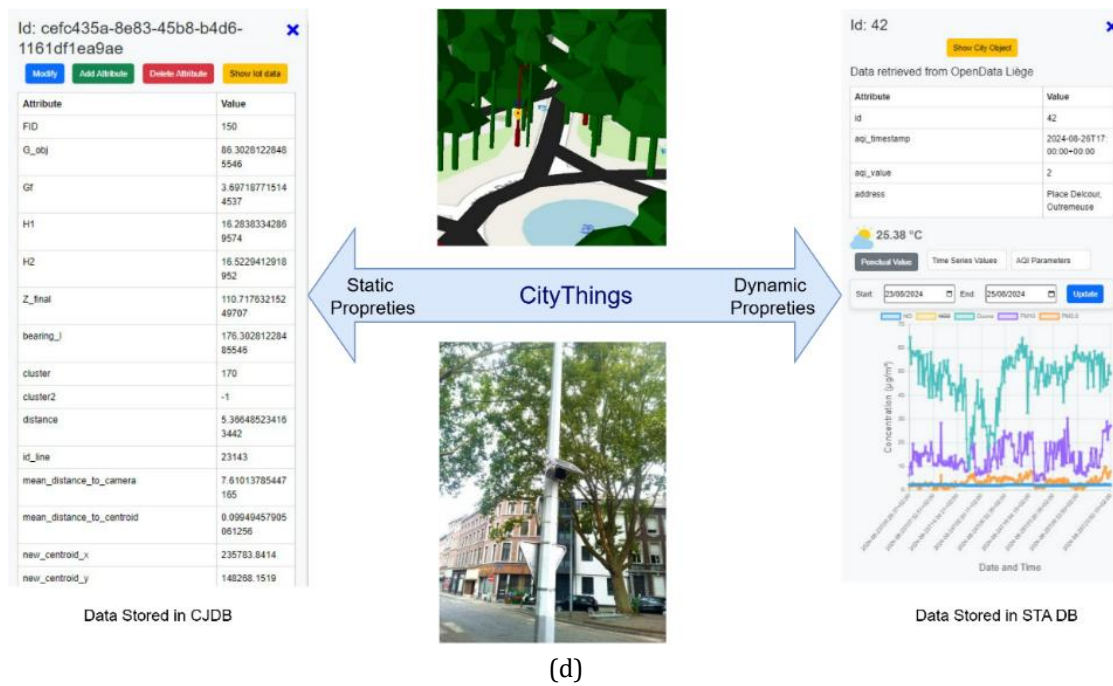


Figure 5.7. The outputs of scenario 2: (a) Level 1- Lamppost with CityJSON v2.0 "Dynamizer" extension, (b) Visualization of "Dynamizer" in City2Twin, (c) Level 2- Integration air quality measurements in CityGML3.0, (d) Level 3- CityThings concept, a bidirectional link between the CityJSON model and the air quality sensors.

5.4.3 Scenario 3: Integrate simulation results into UDT

Used Datasets

In this scenario, we evaluate two different case studies, each with its own 3D city model and simulation workflow. A dataset overview is provided in Table 5.3, and implementation details of both combinations are summarized in Table 5.4.

Table 5.3. Datasets overview, source and description.

Data	Source/Provenance	Description
3D city model - CityJSON	3D reconstruction using Roofer	A 3D city model of Sartilman campus, a former village in the south of the city of Liège. The model has 1667 buildings, 16 Functional Categories, (1918-2025) Construction Years and (1.76-168m) Height Range.
Heat demand	IDEAS Modelica library from KU Leuven within the Dymola simulation platform	JSON files with annual heating and cold demand metrics in MWh, specific heating and cold demand in kWh/m ² /year, and electricity and DHW demands for each building identified by its unique ID.
3D city model - CityGML2.0	HfT_Stuttgart_Campus, Test data repository from SimStadt v2 tool.	8 buildings, 1 functional category, construction years ranging from 1870 to 2016, and building heights ranging from 11.3 m to 29.0 m.

Heat demand	SimStadt v2 ³⁷	Heat Demand Analysis, calculation of the yearly space heating and domestic hot water demands, based on DIN V 18599.
--------------------	---------------------------	---

Table 5.4. Combinations of energy datasets and 3D city models and their implementations.

Combination of implementations	CityGML 2.0+ Simstadt results	CityJSON + IDEAS Modelica results
Energy ADE -Level 1	Yes	CityJSON Energy extension still exploratory
Generics-Level 2	-	Yes
STA -Level 3	-	Yes
Customized architecture – Level 3	-	Yes

Software used and workflow

We implemented a Level 1- CDM workflow that couples building-scale energy simulation with a CityGML-based data model. Heating demand is simulated with SimStadt, producing one consolidated CSV per workflow run at monthly and annual resolution. A Python-based ETL parses this CSV and maps each building's records to the Energy ADE 2.0 Resources module: for every building, we register space-heating as a resource and attach annual values, while generating required identifiers (UUID-based gml:id) and preserving referential links to the source building features. The results are inserted into a 3DCityDB instance extended with the Energy ADE 2.0 schema, exported as CityGML with Energy ADE content, and validated against the schema to ensure syntactic and structural consistency. The core outcome at this stage is data integration; visualization is presented as two viable paths rather than a completed deployment because our UDT platform (City2Twin) currently supports CityJSON only, so rendering Energy ADE content from CityGML requires additional conversion and attribute management steps. Furthermore, native 3D rendering support for Energy ADE content remains limited in common viewers, two established options are: (1) mapping per-building energy attributes to CityGML generic attributes to enable immediate color-mapped rendering in common viewers and to preserve attributes through conversions (e.g., OBJ); and (2) exposing Energy ADE Resources via an OGC WFS, linked to building geometries through gml:id, thereby decoupling attributes from 3D tiles and supporting lightweight, on-demand queries. In this work, we adopt the second approach (OGC WFS). Simulation results are first integrated into 3DCityDB; the results are illustrated in **Figure 5.8**, confirming correct attribute mapping, database insertion, and export. Then we configure the OGC WFS to expose and access Energy ADE 2.0 attributes. The CityGML building model is converted to 3D Tiles using FME, while attribute retrieval remains via WFS; client-side link is achieved through GML IDs. For visualization purposes, a cesium-based web viewer is implemented to query WFS dynamically and visualize energy metrics.

For level 2- DB and Level 3- front-end, we use the same CityJSON model and the same simulation outputs to illustrate their implementations. At level 2- DB, integration is achieved by adapting data to the existing conceptual data model and schema without modifying the model itself. In our case, the CityJSON schema remains unchanged; instead,

³⁷ <https://simstadt.hft-stuttgart.de/>

we feed specific objects using generic attributes that hold the energy metrics. Concretely, we run an ETL pipeline that reads simulation outputs, uses a Python step to inject per-building fields such as HeatDemand (MWh), SpecificHeatDemand (kWh/m²), and a derived EPC_Class into the CityObjects' attributes, and then writes back a schema-valid CityJSON file. This maintains semantic consistency while making the dataset reusable for energy analysis. Because CityJSON is lightweight and easy to parse, these enriched files remain valid CityJSON 2.0 models and can be reused in third-party software as well as in City2Twin. On the platform side, we also updated the client-side data handler so that these generic attributes are recognized and classified at runtime. The platform maps outputs to the platform's fixed energy classes to guarantee correct building classification and filtering, and it also exposes SpecificHeatDemand. This approach preserves the three-tier architecture (CityJSON DB, Flask server, browser client) while enabling fast loading and analysis, with an unchanged core data model and a repeatable ETL that simply adjusts the data to the model, not the other way around.

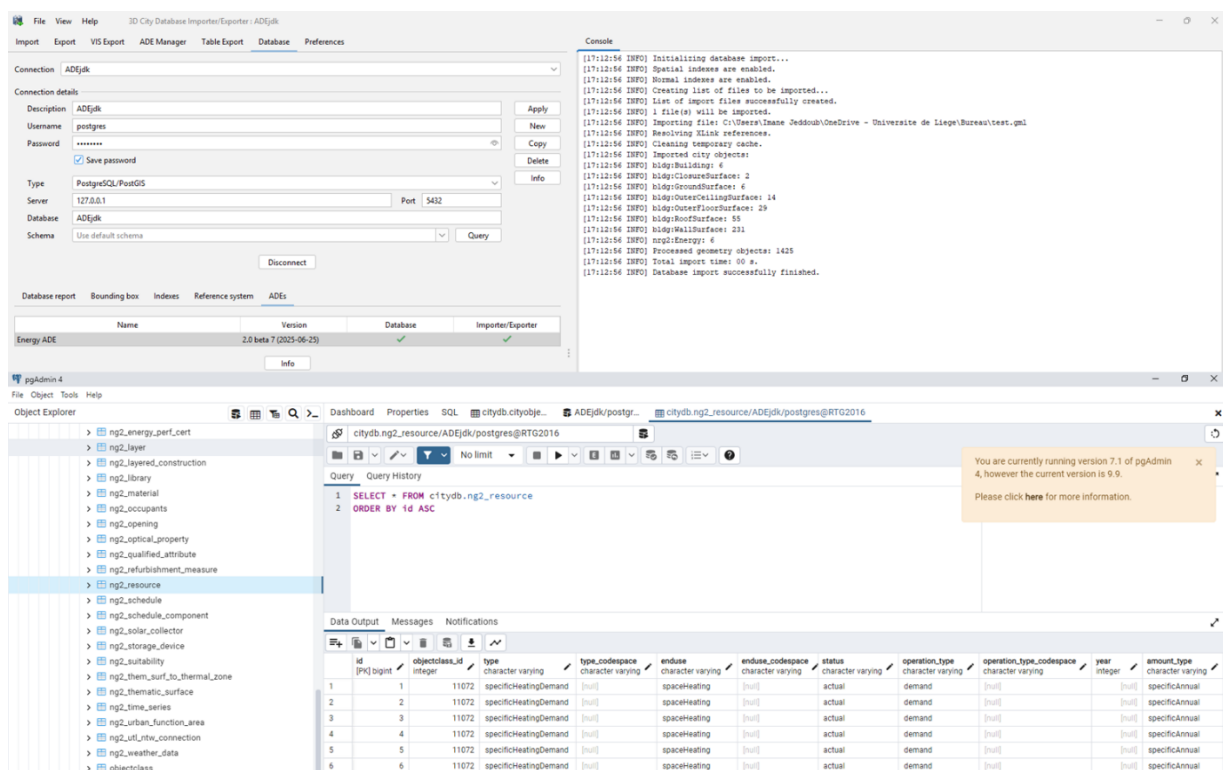


Figure 5.8. 3DCityDB workflow: Energy ADE 2.0 data integration and export.

For level 3- front-end, we adopted two complementary integration workflows to couple building-level energy metrics with a CityJSON model in City2Twin. In the first (level 3-OGC STA standard) workflow, we evaluated the suitability of OGC SensorThings API (STA) to store and manipulate the annual energy attributes. Each building is treated as a unique entity: its identifier becomes properties.building_id on a STA Thing entity, and its semantic GroundSurface footprint is extracted from CityJSON, reprojected from EPSG:3812 to WGS84, and published as the Thing's location. For every simulated energy metric (e.g., Specific Heat Demand, Annual Heat/Cold/Electricity, Domestic Hot Water) we create a dedicated Datastream with appropriate units and post a single annual Observation (i.e., phenomenonTime = simulation year). On the client side, City2Twin client resolves a clicked CityJSON object to its building_id, queries the corresponding Thing (expanding

Datastreams and latest Observations), and writes the retrieved Specific Heat Demand back to the local CityJSON attributes, deriving an EPC label via fixed Belgian thresholds. The EPC_Class is then derived using fixed Belgian thresholds (A+...G) and visualized with a predetermined color palette; legend checkboxes enable interactive, client-side filtering of classes without additional server requests.

The second (Level 3-customized architecture) workflow implements a direct, client-side mapping between simulation outputs and the CityJSON model. Simulation results are provided as a JSON keyed by building_id and loaded in the browser. We inject metrics, such as SpecificHeatDemand (kWh/m²), HeatDemand (MWh), and the derived classes into their corresponding buildings, preserving a one-to-one link via unique identifiers. On load, the client merges these attributes with the 3D geometry and exposes them to the UI for interactive analysis: buildings are color-coded by EPC class, users can toggle scenarios (e.g., standard vs. insulation) and instantly see class migration and percentage savings, and charts update live to show distributions across classes and average heating demand by function. Since the workflow is lightweight and browser-side, it avoids heavy ADE dependencies while remaining standards-friendly, enabling multi-scale filtering (district to building) and “what-if” scenarios. The same mechanism also supports user-provided models and multiple result files as long as IDs match, the integration logic ingests the datasets, enriches the CityJSON on the fly, and renders enriched energy datasets where planning questions (e.g., which typologies to retrofit first, or where envelope upgrades yield the biggest gains) are answered visually and immediately.

Results and discussion

The outputs derived from each integration level differ in structure, persistence, and interactivity, reflecting the integration depth between the building energy simulation results and the UDT framework.

The Level 1 integration delivers a schema-valid CityGML 2.0 dataset enriched with Energy ADE 2.0 (Resources), in which each building carries heating demand values referenced by stable gml:id. The same content persisted in a 3DCityDB instance extended for Energy ADE, enabling reliable storage and SQL-based analysis (e.g., filtering by demand thresholds, aggregations by building function or area, and retrieval of time-series per building when inserted). The export preserves semantic consistency and identifiers, ensuring interoperability and straightforward joins with external UDT layers. For visualization, the cesium-based platform links both the 3D city model and the Energy ADE 2.0 attributes (refer to **Figure 5.9.a**).

At Level 2, the output is a valid CityJSON file enriched with generic energy attributes (see **Figure 5.9.b**). These attributes are embedded within the CityObjects' attribute through the ETL process, ensuring that the dataset remains standard-compliant, reusable, and portable across tools and systems. This approach retains the original schema and guarantees semantic consistency. However, while the integration ensures compatibility and reliability, the resulting dataset remains static, meaning that any new scenario or update requires regeneration through another ETL process.

At Level 3, two complementary workflows generate progressively more dynamic and interactive outputs. The standards-based workflow (Level 3a) uses the OGC STA, implemented through a FROST-Server instance, to produce a web-accessible and

queryable dataset. In this configuration, each building is represented as a “Thing” whose properties include its building_id, geometry, and related energy metrics. Each simulation output (e.g., heat demand, electricity, or domestic hot water) is stored as a Datastream containing corresponding Observations. This structure enables on-demand retrieval, update, and filtering of energy data through REST queries, offering a dynamic connection between the UDT and the BES results (see **Figure 5.9.c**). The STA-based output therefore introduces system-to-system interoperability, allowing future integration of temporal or sensor data for continuous monitoring. However, at the current stage, the implementation uses only annual observations, meaning that the broader capabilities of STA, such as handling continuous time-series data and real-time updates, are not yet fully deployed. Moreover, while the FROST architecture provides flexibility and scalability, it also introduces additional technical trade-off related to server deployment, database maintenance, and API configuration.

The customized architecture (Level 3b) implements different approaches by enabling direct client-side integration between simulation results and the CityJSON model. The output in this case is interactive, it exists only during the user session but allows immediate visualization of different energy scenarios (e.g., standard vs. insulation) (refer to **Figure 5.9.d**). Users can visualize EPC-class migrations between two simulation scenarios, compare heating demand metrics, and explore retrofit impacts through real-time visual feedback. This workflow prioritizes lightweight deployment, interactivity, and responsiveness, making it particularly effective for exploratory analysis or participatory decision-making. However, since the integration is mainly client-side, data persistence and scalability are limited, large datasets may affect browser performance, and results must be exported to maintain the new energy datasets.

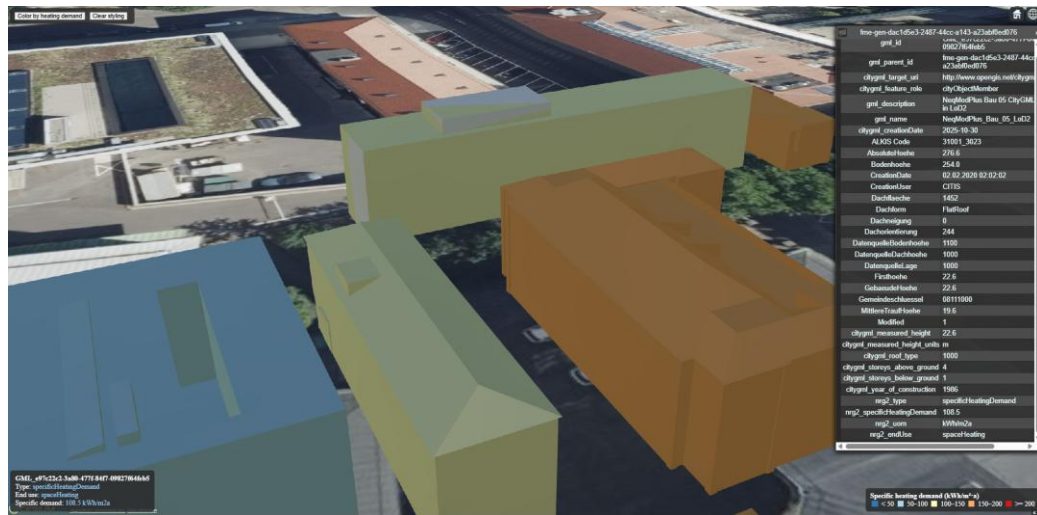
Building on these outputs, the comparative analysis in **Table C.3** highlights the respective pros and cons of each integration. It shows that Level 1 is a semantic driven, consistent and persistent approach. By using a formal ADE, the dataset achieves schema validity, stable identifiers, and explicit temporal structures, which together support rigorous analysis. In practice, our ETL mapped the SimStadt CSV to Energy ADE Resources and stores the results in 3DCityDB, providing a validated CityGML file. The principal limitation is ecosystem readiness, current viewers and web services still lack first-class ADE support, which constrains direct visualization and needs either attribute conversion (e.g., to generic attributes) or a dedicated service layer such WFS. Consequently, Level 1 is optimal for storage, and database queries, while interactive rendering depends on complementary architectures.

As shown in **Table C.3**, Level 2 highlights semantic consistency with comparatively lightweight files, making it well-suited for reproducible platform integration. Our implementation preserved schema-valid CityJSON and injected energy attributes (e.g., HeatDemand, SpecificHeatDemand, EPC_Class) as generic attributes, enabling rapid loading and visualization. The main limitation is that these outputs are static snapshots (i.e., interactivity and scenario updates require regenerating files, and dynamic linkage to external services is not intrinsic to this level).

For Level 3a (FROST implementation), the approach highlights API level interoperability, on-demand access, and robust spatial and semantic linkage as key strengths. In our implementation, STA entities (*Things*, *Datastreams*, and *Observations*) were aligned with

CityJSON building identifiers and locations, enabling clients to retrieve energy metrics and compute EPC classes dynamically. However, these benefits come at the cost of higher implementation complexity and potential performance overhead for API requests. In addition, the semantic representation achievable with STA remains more limited compared to the semantic richness supported by ADE extensions.

For Level 3b (customised client-side architecture), this approach emphasizes lightweight, highly responsive interaction and user-driven analysis. Our client merged per-building JSON metrics with the CityJSON model at runtime, enabling instant scenario switching, class migration visualization, and live charts. The key limitations are client-side memory and performance limits for large scene rendering and a narrower semantic scope than ADE-based representations. Future work will couple the client directly to the simulation model enabling on-the-fly computation of energy outputs and immediate visual feedback. This level maximizes flexibility and interactivity, offering an intuitive way to analyze multiple energy scenarios, though at the cost of persistence and scalability.



(a)



(b)



(c)



(d)

Figure 5.9. The outputs of scenario 3: (a) Level 1- CityGML model with Energy ADEv2.0 visualized in Cesium-based platform, (b) Level 2- 3D CityJSON model using generics, (c) Level 3- OGC STA implementation, linking the CityJSON model with the energy STA service, (d) a lightweight front-end implementation following data mapping and linking using unique ids.

5.5 Discussion, Practical Insights, and Recommendations

This section is organized into two subsections (1) general discussion about the integration levels, (2) practical insights, lessons learned based on our real-world implementations and recommendations for future work. This section synthesizes our conceptual and technical findings on data integration levels.

5.5.1 General discussion: Why do integration levels matter?

In this work, we aim to illustrate and position the integration levels as mandatory step across the UDT lifecycle. Level 1 highlight data consolidation and harmonization with a well specified conceptual Urban Information Model typically by extending CityGML with ADEs or by profiling the base model to the use case. Operating at this level yields schema validity, stable identifiers, and explicit producing analysis-ready datasets that can be validated and reused across tasks, tools, and projects. Compared with Generics, ADEs are formally defined in their own namespace and come with a schema, which reduces ambiguity and naming conflicts and allows more detailed, structured content (including proper time-series). Generics are still useful as a lightweight option, for example in Level 2 snapshots when full ADE support is not available or practical, but they do not offer the same semantic rigor or formal validation expected at Level 1. Level 2 provides schema-valid, platform-ready snapshots (e.g., CityJSON generics) that are lightweight to load and easy to reuse. This level prioritizes portability and rapid delivery while processing the data to meet the data model requirements without altering the schema. Level 3 privileges linkage through standards or lightweight client-side fusion, enabling selective, on-demand access and interactive querying and visualization.

These levels also capture differences in data representation and access mechanisms. Level 1 supports detailed time series embedded in the data model (e.g., CityJSON extension). Retrieval and update pathways are largely determined by the data model design and ingestion workflow. Level 2 delivers compact, cacheable files optimized for rapid loading, sharing, and reuse. Level 3 offers flexible access through standards-based APIs with server-side filtering and access control, or through client-side data fusion at runtime. Update behavior is a central discriminator to understand the role of these data integration levels. In our implementation, Levels 1 and 2 rely primarily on ETL pipelines; Level 3 supports event-driven ingestion and retrieval during periods when no new measurements are available, the API simply returns no additional observations avoiding the need to rebuild files (in our tests, the STA service simply returned no new observations during periods when the sensor was temporarily unavailable due to maintenance). Furthermore, the integration framework is technology-agnostic and transferable beyond the scenarios presented in this work to other datasets, use cases, and contexts, assuming identifiers and semantics are consistently governed and documented.

Although multi-source integration has been discussed in prior work, the growing adoption of UDTs makes explicit mechanisms and levels of integration increasingly necessary. Formalizing levels allows us to answer foundational questions about where consolidation is required, where linkage suffices, how updates propagate, and how these approaches (e.g., CityGML-ADE, CityJSON snapshots, API endpoints) can be designed for reuse across projects. It also clarifies interface responsibilities. For instance, Levels 1-2 typically require viewers or web clients to access, manipulate, and visualize the data. However, Level 3 is itself a client or service architecture that exposes data through interactive views by design. To ensure general understanding across levels, comprehensive documentation and metadata covering identifiers, scenario, simulation assumptions, timestamps, and provenance are indispensable (e.g., in our test using Modelica, we established a set of rules to ensure a seamless and standardized integration process). Moreover, Level 3 emphasizes profiling standards for the intended purpose (e.g., OGC- STA is natively aligned with sensing and can host simulation results with appropriate timestamp and mapping; Dynamizer support well-defined semantics and time series;

CityJSON generics support web-friendly delivery). Finally, cost and organizational constraints are also relevant in such analysis. While our analysis is academic, deployment choices are shaped by resource availability that differs across government and private sectors. Thus, data integration levels guide these choices, which means that consolidation (Level 1) concentrates modelling effort where analytical depth is required; distribution (Level 2) reduces delivery costs for platform integration; linkage (Level 3) reduces the cost of sharing and supports incremental growth. Accordingly, the levels provide a comprehensive basis for designing UDT that balance analytical reliability, interaction, and sustainability while remaining extensible to broader datasets, standards profiles, and institutional contexts. We defined data integration levels according to the lifecycle and the main purposes it serves rather than a practical approach. The central questions concern why data should be integrated and for whom. Non-expert users often prioritize timely and lightweight access, which is typically supported by Levels 2 and 3, whereas domain experts tend to prefer semantically rich and consolidated approaches, as reflected in Level 1. Our objective is not to identify a single most effective level, but rather to make explicit the key factors that guide these choices, including application intent, update strategy, access requirements, cost, and software readiness, and show how they shape interfaces and outputs. Articulating the levels in this way helps avoid the past challenges of generating multiple city-model variants that soon become obsolete, reduce duplicated modelling effort, and offers a reproducible basis for selecting consolidation, distribution, or linkage pathways that align with user needs and organizational constraints.

5.5.2 Lessons learned, practical insights and recommendations

Based on our implementations, we draw the following practical insights about the integration levels:

- **Unique identifiers are the backbone:** Stable and cross-domain IDs (e.g., gml:id, building_id) are essential for lossless joins across BIM updates, simulations, and sensor feeds.
- **Purpose and prior knowledge guide selection:** Understanding the use-case requirements (who, why, and what needs to be created, used and updated) determines the appropriate levels.
- **Data representation and retrieval is a real trade-off:** Level 1 maximizes semantic representation to the city object, whereas Level 3 maximizes retrieval flexibility and distribution; Level 2 balances portability with sufficient semantics for platform use.
- **Software readiness and support matters:** Data only becomes useful with mechanisms for access, visualization, and dissemination. ADE tools/viewers remain limited; STA is mature on the API side but needs careful mappings for simulated (i.e., not sensed) data. Bridging via generic attributes or lightweight transformers keeps workflows operable.
- **Keep workflows format-agnostic:** The levels define the integration logic, not any particular encoding or software tool. We maintained portability across Energy ADE/Dynamizer, CityJSON (with generics), STA/FROST, and client-side JSON by preserving consistent identifiers, and attribute conventions, with clear schema mappings. The framework is therefore not tied to a single format or tool, which explains the diversity of components used in this work.
- **Web performance and usability are critical:** Large scenes and 4D views require tiling/LOD, caching, and attribute retrieval. Our CityJSON workflow improved load

- times and runtime classification (e.g., EPC class mapping) while preserving the core model.
- **Use the integration levels to design sustainable UDTs:** The levels should be considered as a blueprint for where to consolidate, what to distribute, and how to link, minimizing version proliferation and duplicated modelling effort.
- **Data quality is relevant:** Accuracy, completeness, availability, and semantic fidelity determine downstream validity namely for semantic loss during extraction and transformation. As with IFC to CityGML pipelines, real-world datasets often lack complete semantics or use inconsistent conventions, which constrains any “ideal” Level 1 representation. Some features do not map 1:1 and require derived calculations or workflow-specific rules. For IoT, temporal coverage and sensor reliability vary; for simulations, inputs/assumptions may differ by study area. These realities make workflow-dependent availability a practical constraint at all levels.
- **Validation must cover more than schemas:** Files may be schema-valid yet contain geometric/topological errors (e.g., self-intersections, non-manifold solids etc.) introduced during conversion. Proposing tools that allow geometry repair, topology checks, and ID consistency checks alongside XML/JSON schema validation.
- **Standards alignment:** Our implementations adopt current specifications and follow their evolution. This improves interoperability and tool support and strengthens long-term maintainability and reproducibility.
- **Plan clear update pathways:** Align update methods with the chosen level.

5.6 Conclusion

Data integration is a crucial step throughout the entire UDT lifecycle. It plays a key role from the creation phase, where geospatial data is integrated and aligned with urban information models, to the use phase, where it supports application-specific needs, as well as during updates and change detection processes. While the data integration challenges are strengthened in many UDT implementations, this work is the first to address them using a dual approach (conceptual and technical) and proposes the levels of integration as a comprehensive framework and technical reference of multisource data integration for UDT. This work considers real-world use cases and various data characteristics to illustrate the integration levels using systematic workflow. By following the provided workflows, UDT practitioners can test the data integration levels using their own datasets and draw conclusions on the integration level suitable for their use case. Furthermore, this analysis of integration levels provides a benchmarking and assessment framework that enables stakeholders to determine the UDT lifecycle stage, the levels implemented in each phase, and the levels that may be adopted in the future to address specific needs, with a particular focus on updates. Indeed, the update and maintenance phase is strongly coupled with the integration levels, and each level has its own advantages and trade-offs, and these can only be evaluated and validated when new data becomes available. This means that, although integration is a fundamental step prior to any UDT implementations, it is recommended to build on our findings to decide earlier where to consolidate (Level 1), what to distribute (Level 2), and what to link via standards-based services or client integration (Level 3), with clear conventional rules for identifiers, attributes, and semantics. Furthermore, adopt hybrid configurations if needed to align to the UDT use case requirements. In addition, planning maintenance pathways explicitly is mandatory to sustain the UDT infrastructure. It is also recommended to keep workflow formats

agnostic, match standards to the use case (ADE/Dynamizer for structured content; SensorThings and WFS for queryable access; CityJSON+Generics for web delivery), and validate beyond schemas (i.e., geometric and topological checks). Resource planning should cover the client and visualization layer for Level 1 and Level 2 and operational services for Level 3. Moreover, lightweight and open standards that support data linkage can lower distribution costs and improve reuse. It is also relevant to highlight that there is not just one possible level for given data or use cases; different configurations (i.e., combinations of the levels) may be appropriate. Our premise is to recognize these levels and their relevance for UDT, as well as to explicitly set conceptual and technical guidelines to support wider adoption.

In future work, our perspective is to extend tests to additional datasets and tools, explore an IFC extension for CityJSON compatible with our UDT architecture, and examine centralized and federated twins across these integration levels. Integration design should consider governance architecture (e.g., a single, centralized twin (strong consolidation)) compared to a federation of domain twins linked via identifiers and APIs. Our levels accommodate both patterns (consolidate where necessary; link where sufficient). Similar to levels of interoperability, we aim to formalize the levels of integration defined through standards profiles to offer a clearer pathway for adoption and benchmarking.

CHAPTER 6: Conclusions and Future Perspectives

The aim of this chapter is to provide a synthesis of the main findings and contributions of this thesis. It revisits the objectives and research questions introduced in chapter 1 and summarizes how the conceptual work (chapters 2 and 3) and the practical developments (chapters 4 and 5) contribute to the understanding and implementation of data integration within Urban Digital Twins. This chapter brings the key results, highlights the main contributions and remaining challenges while outlining perspectives that logically follow from the work.

This thesis is rooted in a key observation from both literature and practice. Although UDTs are increasingly positioned as a promising concept for cities, data integration is considered both a fundamental requirement and a major challenge, which still hinders full operational implementation. At the same time, the need for flexible and web-friendly solutions is growing, while the geoinformation community continues to rely on standardized semantic urban models. Bridging these two perspectives (i.e., semantic rigor and flexibility) has guided the research and the design choices made in this thesis.

Overall, this thesis aims to address, to some extent, how UDTs can be designed to integrate heterogeneous data while remaining flexible, lightweight, and maintainable across diverse urban applications. To this end, the thesis is structured around a set of conceptual and technical research questions. In the following sections, each research question is revisited and the main findings are summarized, before positioning the overall contributions and suggesting perspectives for future work.

6.1 Findings and contributions

6.1.1 Research questions

The main research question of this thesis is approached progressively through six conceptual and technical research questions.

RQ1. How should a UDT be defined with respect to data integration and interoperability requirements, and to what extent does real-time interaction shape this definition?

This thesis confirms that the term “Urban Digital Twins” is used in the literature and in practice in different ways which increase the divergence in terms of implementations and initiatives. Through a review of definitions and a comparative analysis with related concepts namely 3D city models, City Information Modeling, and Spatial Data Infrastructures, we show that UDTs do not represent a single form, output, or systems of digital models. Instead, they are positioned along a spectrum of maturity, scope and purposes they serve.

In this thesis, UDTs are considered as data-driven infrastructures that build upon semantic 3D city models but go beyond static representations by explicitly targeting continuous enrichment, integration of heterogeneous data sources that support analysis, simulation, and decision processes, and providing update mechanisms aligned with the UDT lifecycle. Most importantly, our findings question the common assumption that real-time interaction should be considered a strict requirement. While real-time and near real-time synchronization is often presented as a key feature, we show that many initiatives

operate with asynchronous update cycles and that the relevant requirements is not “real-time” per se, but an appropriate rate of synchronization aligned with the use case requirements and data availability.

This work therefore provides pragmatic clarification. Rather than deciding whether an initiative is simply a “twin” or “not a twin”, it is more useful to assess it in terms of its maturity and data integration perspectives. This perspective also avoids overpromising capabilities and supports implementations that can evolve over time.

RQ2. How is data integration addressed within the emerging concept of UDTs? Does the UDT paradigm introduce new data integration challenges compared to previous geospatial and urban modeling approaches?

This thesis shows that data integration is widely recognized as a major challenge, yet it is rarely addressed in a systematic and explicit manner. Current UDT implementations integrate data through various approaches that are strongly driven by use cases, available datasets, and organizational constraints. As a result, integration is frequently addressed during initial setup only, rather than being managed continuously across the lifecycle.

In addition, this thesis shows that the UDT paradigm introduces new data integration challenges, given that most UDT implementations rely on existing standards that are not yet fully mature to meet the UDT requirements. Furthermore, urban data are heterogeneous not only in terms of formats and semantics, but also with respect to their spatial and temporal scales, as well as their organizational ownership. Beyond these aspects, integration in UDTs also depends on the integrability of the available data, that is, whether datasets can be combined in practice given their structure, quality, and modeling assumptions, and whether the effort needed for harmonization is justified. Differences in geometric representations and modeling paradigms across domains, as well as the level of detail needed to support specific applications, further increase the complexity of integration. This complexity is inherent to the fact that a UDT, like any digital model, is an abstraction of reality, built for specific purposes and constrained by scale, level of detail, and temporal resolution. As such, a strict 1:1 digital representation of the city is neither realistic nor necessary to support urban applications. These dimensions make it more difficult to maintain data consistency, support reuse and automation, and manage long-term updates within a UDT.

Overall, this thesis highlights that while data integration challenges increase in the context of UDTs, the UDT paradigm also offers an opportunity to better organize and structure integration efforts by fostering collaboration between stakeholders and promoting data reuse within a shared digital representation of the city.

RQ3. How can data integration be defined and formalized according to different approaches across the UDT lifecycle (creation, use, and update)?

The main conceptual contribution of this thesis is the Levels of Integration (LoI) framework, which formalizes data integration according to where it takes place in the UDT infrastructure and how it is framed to the different stages of the UDT lifecycle. The framework proposes and distinguishes three main levels of integration, (i) the conceptual data model, (ii) the database level, and (iii) the front-end level.

These levels reflect different ways of incorporating multisource data to a UDT, ranging from schema-level extensions and semantic alignment to database-level coupling, and lightweight integration at the front-end layer.

It is important to note that these levels do not represent a hierarchy of more or less effective solutions. Rather, they reflect different degrees of interoperability and coupling between data, models and the UDT, depending on the intended use and context.

By relating these integration levels to the UDT lifecycle phases (creation, use, and update), this thesis shows that data integration requirements evolve over time. Integration during the creation phase is mainly associated with Level 1 integration, where data are aligned at the conceptual level. During the use phase, integration is more often supported through Level 2 and Level 3 integration, enabling data to be accessed and combined for different applications. During the update phase, integration becomes more complex and typically requires stronger coupling mechanisms, as it must address data consistency, synchronization, and change management over time.

Overall, the Levels of Integration framework provides a structured way to study, compare, and discuss data integration approaches in UDTs, while making explicit their scope and limitations. It supports a better understanding of how different integration strategies can be selected and combined depending on the purposes and constraints of a given UDT.

RQ4. How can an Urban Digital Twin framework be developed to support different levels of data integration in practice?

This thesis investigates how an urban digital twin framework can be developed to support different levels of data integration in practice. The development of City2Twin was motivated by two main objectives. First, it aims to challenge the current discourse on UDT implementation, which is often based on 3D CityDB and CesiumJS approaches. Second, it intends to document the implementation process step by step, since such development approaches are often not described in detail in the existing literature, while data integration is mainly framed as a challenge rather than explicitly documented as an implementation process.

The development and evaluation of City2Twin show that supporting data integration in practice is not related only to selecting a particular set of tools. Rather, it requires a coherent framework that defines how urban data are represented, stored, accessed, updated, and made available through an interactive client. Such a framework must also remain extensible and maintainable over time so that it can accommodate different forms and levels of integration across a variety of urban datasets and use cases.

Based on the literature review and the implementation of City2Twin, several key framework requirements are identified. These include: a validation step to ensure that ingested city models are consistent and usable; a flexible data model capable of representing urban objects and their semantic properties while remaining extensible; a storage solution that supports spatial queries, partial retrieval, and incremental updates; mechanisms for handling both static and dynamic urban data; a backend application layer

that manages data access and updates; and a client interface that enables users to inspect and interact with the integrated data. From this perspective, City2Twin should not be understood only as a technical prototype but as a technical framework that illustrates how urban digital twins can be developed to support data integration in practice. Its contribution lies not only in the resulting platform but also in making explicit the development steps and design decisions that are often overlooked in the state of the art. The implementation shows that a lightweight solution based on CityJSON and CJDB can support interactive web-based applications while enabling spatial filtering, partial retrieval, and interactive attribute updates with reasonable response times. At the same time, it highlights that the effectiveness of such a framework depends on practical design choices, including export workflows, indexing strategies, and effective handling of concurrent requests. Overall, City2Twin demonstrates that developing an Urban Digital Twin framework for data integration requires aligning conceptual structure, technical implementation, and practical usability.

RQ5. How can different levels of integration be implemented across various UDT use cases?

This thesis shows that implementing different levels of data integration in UDTs goes beyond selecting a standard for 3D city models. It requires defining concrete integration workflows that are adapted to the nature of the data, the intended use case, and the expected evolution of the system. Chapter 5 demonstrates this through the implementation of several representative integration scenarios, covering static, dynamic, and detailed models (e.g., IFC models).

One of the proposed scenarios focuses on the integration of dynamic data, which represents a key challenge for UDTs. Motivated by the CityGML 3.0 Dynamizer module and its role in linking time series to city object attributes, this thesis develops and validates a CityJSON-compatible extension inspired by the Dynamizer concept and integrates it within the City2Twin platform. The extension was upgraded and validated for CityJSON 2.0 and implemented as a complete workflow, including a server-side service that retrieves observations from an OGC SensorThings API backend and writes both punctual values and time series into the CityJSON structure, while a client-side visualization layer enables users to explore static and dynamic representations of the same city object.

Beyond the technical implementation, this contribution is also methodological. It shows how a CityJSON-based UDT can support dynamic data integration while preserving a lightweight data structure. At the same time, the experiments highlight that tool support for dynamic extensions remains uneven, and that integration choices must balance expressiveness, performance, and maintainability.

The thesis further shows that different integration levels lead to different outputs and usage possibilities. Conceptual-level integration supports semantic alignment and standardized representations but remains constrained by schema complexity and limited tool support. Database-level integration enables structured storage and query while requiring compromises in expressiveness. Front-end-level integration supports flexible

visualization and interaction, particularly for dynamic data, but relies on coupling with external services. In practice, UDT implementations rely on combinations of these levels depending on the data type and the use case.

The energy domain illustrates the need for hybrid integration strategies. While CityJSON extension mechanisms allow the addition of new CityObjects and attributes, they remain limited when representing complex hierarchical structures and non-CityObject elements, which are common in energy models and time-series representations. Existing work on mapping the CityGML Energy ADE to CityJSON shows that such translations cannot be fully direct due to structural differences between XML and JSON, simplifications driven by use-case requirements, and limitations of the CityJSON extension mechanism. As a result, CityJSON-based UDT workflows may still need to rely on CityGML Energy ADEs, WFS-based services, or hybrid representations for energy-related use cases until CityJSON energy extensions mature further.

Finally, this thesis also acknowledges its limitations. The implementation focuses on a limited number of representative integration scenarios; each illustrating specific integration levels and use cases. While these scenarios demonstrate the feasibility and implications of different integration strategies, they do not exhaustively explore all possible combinations of integration levels. Further work is required to study more complex configurations, including mixed strategies where multiple integration levels are combined simultaneously within the same use case.

Overall, this thesis demonstrates that different levels of data integration are implemented in practice by aligning integration workflows with the nature of the data and the intended UDT use case. The integration levels are not alternative solutions competing with each other, but complementary approaches that can coexist within the same UDT. By explicitly distinguishing these levels and illustrating their concrete implementations, this work provides practical guidance for selecting and combining integration strategies according to operational needs, technical constraints, and desired outputs.

RQ6. How can flexible and lightweight UDT architecture support, illustrate and operationalize different data integration levels?

This thesis addresses this question through the design and implementation of City2Twin, developed in chapters 4 and 5. City2Twin demonstrates that flexibility and interoperability can be combined within a UDT, provided that responsibilities are clearly separated through modular and lightweight architecture.

At its core, City2Twin relies on CityJSON as a lightweight semantic 3D city model, stored in CJDB and exposed through web services that allow direct consumption by client applications. Static urban data and attributes are managed directly within CityJSON, while dynamic and domain-specific data are handled through complementary services, such as an OGC SensorThings API, and linked to the city model through explicit identifiers and integration logic. This design enables multiple data integration levels to coexist within the same platform, including conceptual extensions, database-level, and front-end level integration and visualization.

Most of the data integration workflows presented in this thesis are implemented and evaluated within City2Twin. By supporting different integration levels simultaneously, the platform illustrates that integration levels are not intrinsic properties of datasets, but operational choices that depend on data characteristics, user expertise, and intended use. Lightweight and partial integration strategies are well suited for exploratory use cases and non-expert users, where timely access and low interaction cost are prioritized. In contrast, higher integration levels, involving stronger semantic alignment and strict consistency, are more appropriate for expert users and long-term analytical tasks (for instance, EnergyADE).

A key contribution concerns the integration of dynamic data through the adaptation of the Dynamizer concept. This thesis demonstrates how similar principles (Dynamizer) can be applied in a CityJSON-based environment, despite the lack of full native support. The resulting implementation shows that dynamic data can be integrated and visualized without overloading the core city model, while also highlighting current limitations in tool support and standard maturity, particularly for complex temporal and energy-related data.

At the same time, the thesis makes explicit that increased flexibility introduces trade-offs. As integration becomes more distributed across services and applications, the importance of validation pipelines, synchronization mechanisms, and clearly defined update rules increases. These aspects become critical as UDTs evolve toward more federated, system-of-systems architectures. From this perspective, the contribution of City2Twin is not limited to providing a working platform but lies in illustrating through concrete implementation, where integration is straightforward, where it is constrained by current standards and tools, and where hybrid solutions remain necessary.

Overall, this thesis shows that flexible and lightweight UDT architecture can effectively support, illustrate, and operate different data integration levels by relying on modular components, standardized interfaces, and selective coupling strategies. Rather than aiming for a fully centralized or integrated system, City2Twin demonstrates that flexibility and lightweight design are key to accommodating heterogeneous data, evolving requirements, and diverse UDT use cases, while maintaining semantic rigor where it matters.

6.1.2 Summary of the main contributions of the thesis

By combining a conceptual framework with operational implementation, this thesis contributes to the UDT field at three complementary levels.

Conceptual contribution: We clarified the UDT concept through a systematic analysis of definitions and implementations, and we proposed a lifecycle-based framework of data integration. This supports a more realistic positioning of UDT initiatives and reduces ambiguity between conceptual discourse and practice.

Methodological contribution: We structured data integration approaches through the Levels of Integration (LoI) framework and demonstrated its relevance through the cross-analysis of existing initiatives and reproducible implementation scenarios. The framework provides a structured basis for comparing integration strategies and planning integration across the creation, use, and update phases of Urban Digital Twins.

This contribution is further developed through a two-step decision-support framework. First, **Figure 6.1** presents a framework for deriving suitable semantic and geometric integration levels. Organized around these two complementary dimensions (i.e., semantic and geometric), it helps determine whether integration is best addressed through Level 1, Level 2, or Level 3. When the requirements cannot be satisfied directly, the framework allows an iterative re-evaluation of the integration requirements. If no suitable refinement is possible, integration is not recommended under the current conditions.

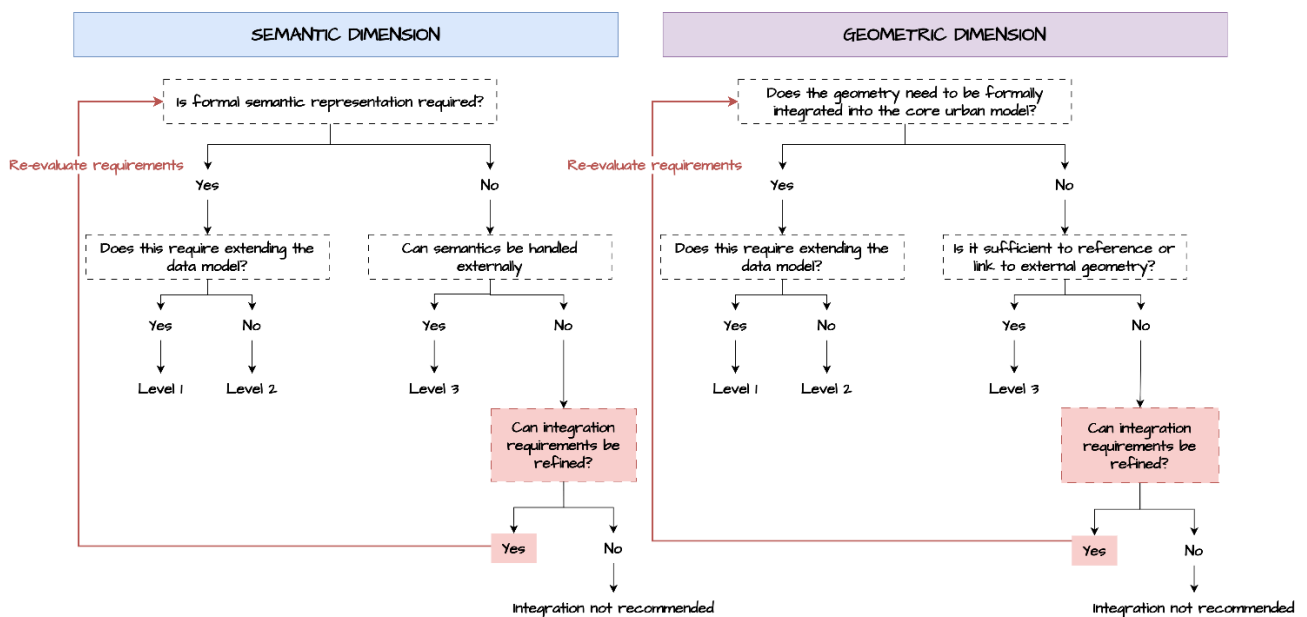


Figure 6.1. Decision-support framework for deriving semantic and geometric levels of integration.

Second, **Figure 6.2** shows how the semantic level (SL) and geometric level (GL) are combined to define the final integration strategy. When both dimensions lead to the same level, that level is adopted directly. When they differ, a hybrid strategy is considered first. If hybrid integration is not feasible, the final choice is guided by contextual considerations, including the lifecycle stage, the target application, the required degree of interoperability, and maintainability and deployment constraints.

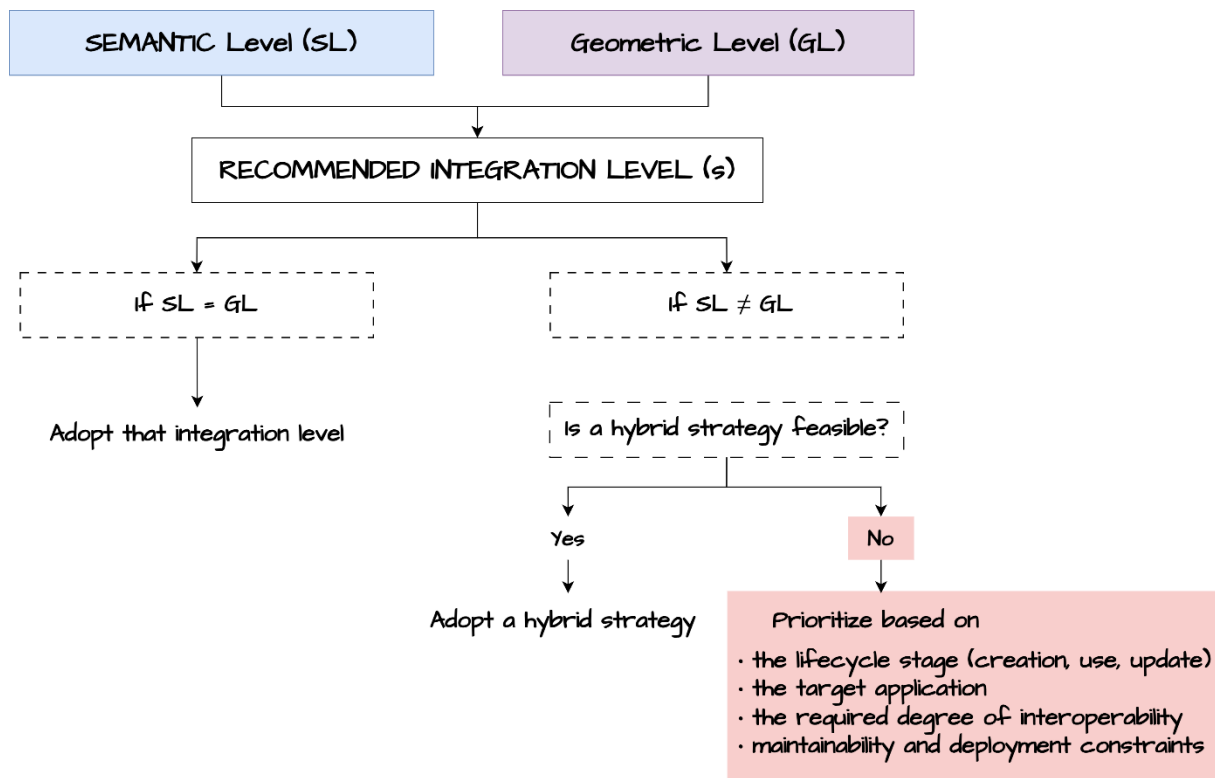


Figure 6.2. Framework for selecting the recommended integration strategy based on semantic and geometric levels of integration.

Overall, this decision-support framework operationalizes the concept of LoI by translating the conceptual and implementation-oriented findings of this thesis into a practical tool for supporting Urban Digital Twin design, implementation, and evolution across different use cases and lifecycle stages. It should be understood as a decision aid rather than as a rigid or exhaustive classification mechanism.

Technical contribution: We developed City2Twin as an open and lightweight UDT platform and documented concrete architectural choices that operationalize different integration levels. This includes adopting CityJSON and CJDB for direct and web-friendly visualization workflows; integrating dynamic IoT data via the OGC SensorThings API; and contributing to the CityJSON ecosystem through the development and validation of a CityJSON-compatible Dynamizer extension for time-varying properties. We also addressed consistency limitations in CJDB-based workflows through custom update and validation logic. Finally, we highlighted domain-specific constraints, notably for energy-related data, where CityGML ADEs and hybrid approaches remain relevant given the current maturity of CityJSON energy extensions.

Overall, this work demonstrates a practical pathway for lightweight UDT implementation based on CityJSON, from data integration to storage, visualization, and analysis.

6.2 Research extensions

While this thesis provides both a conceptual framework and an operational implementation, several limitations must be acknowledged, which also point to natural directions for future research.

First, although City2Twin demonstrates how a lightweight UDT can be designed and implemented, it does not aim to provide a universal or “one-size-fits-all” digital twin solution. Urban Digital Twins are inherently contextual and depend on data availability, technical requirements, and evolving stakeholder needs.

The proposed architecture should therefore be understood as a reference and experimental platform that illustrates the steps and design choices involved in UDT implementation, and that can serve as a basis for further developments toward production deployment.

Second, the maturity of standards and tools remains uneven. While CityGML provides established mechanisms for domain extensions through ADEs, tools around CityGML 3.0 are still evolving. Similarly, although CityJSON offers a lightweight and developer-friendly alternative, its extension ecosystem for complex thematic domains (notably energy) is still under development. As a result, several use cases currently require hybrid solutions that combine CityJSON workflows with CityGML ADEs, WFS-based services, or external APIs.

Third, long-term update and consistency management remain challenging in practice. This thesis shows that flexible and lightweight architectures shift part of the responsibility for consistency away from rigid database schemas toward explicit update pipelines, validation rules, and application-level logic. While such mechanisms were implemented for selected scenarios in City2Twin, further work is needed to generalize them and assess their robustness over longer time periods and across larger and more diverse datasets.

Fourth, the server-side architecture and deployment choices introduce additional limitations. City2Twin currently relies on a Flask-based backend, which is well suited for prototyping, experimentation, and small-scale deployments. However, moving toward production of the UDTs raises challenges related to scalability, concurrency, and security. Supporting large city models, high-frequency sensor streams, and multiple concurrent users would require further investigation into alternative server architectures, load balancing strategies, and deployment environments. These aspects were outside the scope of this thesis but are critical for the operationalization of UDT platforms in real-world settings.

Several research directions represent a logical continuation of this work.

(1) Strengthening data integration across the full UDT lifecycle.

Future work could focus on developing update workflows that preserve semantic

consistency while supporting incremental data ingestion, schema evolution, and provenance tracking. The Levels of Integration framework could also be extended with maturity and governance dimensions (e.g., data quality, access modalities, control, licensing) to better support federated UDT deployments.

(2) Advancing CityJSON extensions and open standards adoption.

The Dynamizer extension illustrates the potential of CityJSON for handling dynamic properties, but further efforts are needed to strengthen standardization, improve tool support, and expand its applicability to additional dynamic use cases such as mobility. Similarly, continued development of the CityJSON Energy Extension could reduce reliance on hybrid approaches if future versions better support complex structures and time series. Another promising direction is the exploration of IFC-to-CityJSON extension pathways to improve BIM-GIS interoperability.

(3) Consistency, versioning, and update management in lightweight databases.

The limitations observed with CJDB regarding consistency and updates highlight the need for systematic approaches to validation, versioning, and synchronization that remain compatible with lightweight storage and direct visualization workflows. For operational use at scale, this also requires studying how indexing and update patterns interact, since interactive updates and partial retrieval depend on both correctness checks and efficient query execution.

(4) Scalability and production deployment of UDT platforms.

Future work should investigate the production deployment of lightweight UDT platforms, including alternative backend frameworks, containerized deployments, cloud-based infrastructures, and performance benchmarking. This would allow City2Twin-like architectures to be assessed under realistic operational constraints, such as large data volumes, continuous updates, and multi-user access.

(5) Profile-driven, knowledge-graph semantic integration for federated UDTs.

Future work will develop a profile-driven integration approach where profiles make modelling choices and semantic assumptions explicit (vocabularies, alignments, quality constraints, aggregation rules). These profiles will guide mappings into a knowledge graph that supports ingestion, transformation, and alignment from the start. This reduces bespoke integration logic and improves reproducibility and reuse across federated UDTs.

(6) Toward federated and system-of-systems UDTs.

As UDTs evolve beyond isolated platforms, integration strategies will increasingly rely on standardized APIs, modular services, and dataspaces. The Levels of Integration framework provides a basis to reason about where integration takes place in such ecosystems, but further empirical evaluation is needed in large-scale, multi-stakeholder contexts.

(7) Broader validation, benchmarks, and dissemination.

Applying the Levels of Integration framework to a wider range of urban contexts, different scales, governance models, and thematic focuses would help assess its robustness and generalizability. Developing open benchmark datasets (i.e., integrable datasets) and reproducible workflows could further support comparison between different integration strategies, databases, and UDT architecture. Finally, producing open-source documentation would facilitate knowledge transfer and adoption by both researchers and practitioners.

In conclusion, this thesis argues that progress toward operational Urban Digital Twins requires both conceptual clarity and transparent technical implementation. By formalizing data integration levels across the UDT lifecycle and by providing a lightweight reference platform that operationalizes these principles, this work contributes to narrowing the gap between what UDTs promise in theory and what can be implemented, deployed, and maintained in practice.

APPENDIX A

Below is a summarizing table of the DT initiatives realized through multidisciplinary work between universities, research institutes and public and private organizations according to the categories listed previously.

Table A.1. Basic information of Digital Twin initiatives whether it is called (City-Urban-spatial) Digital Twin: scale, data sources, city models, level of details, purposes, visualization platforms, simulations use cases, method, frequency of updates and maturity level.

CDT	Scale	Data sources	City models	LoD	Purposes	Visualization platforms	Simulations experiences	Update method and frequency	Status
<u>Helsinki 3D+ Kalasatama</u>	City-District	Oblique photogrammetry Aerial LiDAR Point cloud	City Information Model-3D CityGML model (Buildings flat-roofed and with differentiated roof structures Bridges, tunnels, waterbodies, terrain)	LoD1-LoD2	3D design, urban planning for climate change and development Energy consumption Virtual tourism Life cycle processes and planning, testing, application and services	Semantic 3D city model: VirtualCitySystems Cesium Mesh model: Web and VR interfaces	Semantic 3D city model: Urban planning and Helsinki Energy Atlas, solar power analysis, flood assessments, noise calculations. Mesh model: Tourism	Method: Open-source BIM models	Operational
<u>Espoo</u>	City	3D city model database	Mesh model 3D CityGML Model (Textured building, generics, city	From LoD0-LoD3	Urban construction, planning	Espoo Map Service		Method: Unique attribute (release date) shows a date when an object has been updated in interface services.	Operational

			furnitures, water body, transportation, vegetation, land use and relief) 3D city model with point cloud Hybrid model		Visualization of city objects from above and under ground			3DCM is maintained in Trimble Locus. Frequency: 3 months	
Vienna	City	GIS geodata inventory and data from specialized applications from departments	Digital GeoTwin Semantics 3D geo-objects		a living virtual city replica allowing the monitoring of city existing processes, the generation of new information, the scenarios simulations of planning Allow a better data-driven decision support	VirtualCityMap	Linking the DT with further information (census, economic, energy consumption, maintenance management data) 3D planning data Simulations (solar potential, flood scenarios, disruptive event).	Under development Towards a local urban digital twin	
<u>Rotterdam 3D</u>	City	LIDAR BAG & BGT Aerial photo	Above and underground infrastructures BIM models and 3D city models.	LoD1- LoD2	Climate adaptation, viewsheds and energy performance of buildings	VirtualCitySystems, ESRI, IMAGEM and UNITY use the model as input for analytics and visualizations.	Generic scalable data sources: energy savings & solar potential asset management of	Buildings are updated annually/biennially. Objects in the public space and	Operational

		BGT (Basic Registration Large-scale Topography) BAG (Basic Registration Addresses and Buildings) BOR (Basic Registration Public Space) LVZK (Pipe Collector Card)	Buildings – Terrain – Trees (roots) – Lampposts – Cables and pipelines Storage tanks as a theme to the 3D city model		Integration of the hydrodynamic city model with the 3D DT		the subsurface infrastructure, urban flooding applications GIS analyses Saving potential, green roof potential, solar potential	the pipes are updated monthly from the management system. Terrain model is updated every two years.
<u>Zurich 3-4D</u> (Schrotter and Hürzeler, 2020)	City	Spatial data infrastructure, geodata portal	Buildings, trees, forests and bridges Over 50,000 buildings in various LoD Walls, bridges.	LoD0- LoD1- LoD2	Urban planning in climate change. Urban spatial data infrastructure.	Web application, geoportal Virtual Zurich Zurich 4D.	Visualization air pollution, and construction projects. Visibility, solar potential analyses, noise propagation, flood simulations, shadow calculations mobile phone radiation Different model version, historical models using a time slider	Operational
<u>Virtual</u>	City-Scale	GIS data, Aerial mapping, mobile street mapping of all public roads	Core datasets: digital terrain model, 3D city models (buildings,	Several LoDs	Virtual experimentation and test-bedding planning.	3DEXPERIENCE City Dassault Systèmes	Solar energy production demographics, climate and traffic data.	Operational

		LiDAR and imagery data. Orthophotos CityGML used for vector models and surfaces	roads, coastline, airspace, underground asset and 3D geology), vegetation, cadaster, land use, waterbodies, point cloud, reality mesh, BIM.	urban planning, efficient energy consumption climate, population dynamics.		Installing green roofs Simulation scenarios to adapt the regional temperature	
<u>Amsterdam 3D</u> (Ávila Eça de Matos et al., 2022)	City	3D BAG	Buildings; roads; vegetation; underground parts (pipelines, cables);	City planning	Unity3D	Still manually updated	Operational
<u>Digital twin Munich</u> (Hijazi et al., 2022)	City	Aerial surveys of the entire urban area, 3D point cloud mobile mapping campaigns supplemented by GPS measurements and drone recordings	3D CityGML model	Climate neutral smart cities.	Urban data platform based on the OGC standards	Traffic and Urban expansion simulation based on urban dynamics, 3D city models.	Under improvement
<u>Rennes 3D</u>	City-District	3D model demographic data relating to mobility, health,		Tackling city complexities by involving all parties through	Dassault Systèmes 3DEXPERIENCE platform	Subway planning Building energy map based on demographic,	Operational

		energy, vegetation		a systemic approach.	Platform on the cloud	land and energy data, Solar map, to fit photovoltaic panels Consultation on a tram/bus project an interface on 5G which gathers information on equipment and on exposure in measured values.	
<u>Virtual Gothenburg</u>	City	Parametric, or procedural modeling	Buildings, streets, lampposts and trees plantations and forests, information describing the city and its objects must also be added. Materials and textures are added	Urban planning Urbanization growth Segregation of the city into ten districts with varying socio- economic conditions Climate change affecting the sea level and posing risk of flooding.	Unreal engine Visualization (v4.27) with several datasets from Gothenburg's open data portal City Engine	Scenarios for future projects and traffic simulations Self-driving vehicles Sun and shadow studies Noise and sound and air quality Challenges with torrential rain and segregation .	Operational
Herrenberg (Dembski et al., 2020)	City	Existing geographic data and information and 3D laser scanning data.	3D city model.	Easily accessible routes for elderly people Potential for solar energy production	COVISE (Collaborative Visualization and Simulation Environment)	Mobility, transport and air quality Co-creation Pedestrian movement and	Prototype Operational

		Data from mobile app and space syntax) Movement traces of bicycles (From GPS data) Movement traces of pedestrian (From GPS data) Mobile app)			Suitable routes for drivers. Effects of new constructions and installations by analyzing ambient temperatures and sunlight.		stationary activities Spatial network analysis Air pollution sensors	
<u>Sofia- Bulgaria</u> <u>GATE</u> (Dimitrov and Petrova-Antonova, 2021; Hristov et al., 2022)	District	Sofia Municipality Footprints of buildings, floors number, function, Addresses points and locality DTM-DSM	Buildings and vegetations, CityGML2.0. Attributes (address information). Integrating buildings and terrain model	LoD1	Create a CDT platform for urban management, experimentation. Enhancement of urban processes throughout the entire lifecycle of the urban environment.	Web application. Shadowing buildings according to their attributes. Cesium ion (hosting of the 2D model) and CesiumJS (visualization) QGIS for analysis	Parametric urban planning Air pollution analysis and simulation 15-minutes walkable cities	Operational Proof of concept/Prototype
<u>DUET</u> (Raes et al., 2021)	Cities Athens, Pilsen, Flanders	Municipal GIS data (terrain, trees location, 3D building models etc.), public transport data, including positions of stops and moving vehicles, wider traffic	local 3D city model	LoD2	Virtual city replicas to address the complex interrelation between systems (traffic, noise, air quality).	Virtual City Systems	Traffic data with data on air and noise pollution used to create scenarios for green routing and traffic monitoring within the city.	Operational Proof of concept

		data, pollution data and noise data. City data (street networks, elevation models, IoT sensor data)		Understand effects of potential change. Policy-Ready-Data-as-a-Service.				
Fishermans Bend Digital Twin (Chehrehbargh, 2022)	District-multi-scale from building to city	Massive 3D datasets 2D and 3D data 3D data formats (CityGML, BIM, 3DTiles, DEM)	Buildings; Roads; Underground parts (pipelines, cables). High resolution photo mesh 3D CityGML model	Test the innovation and emerging technologies for better planning and land administration. Planning analysis tool (PedDesign tool enables analysis of pedestrian movement. BIM based 3D Cadaster query and visualization.	a composable, customizable and community-oriented system architecture. Built on a web-GIS and use Cesium and other mediums (e.g., Matterport) for visualization	quantitative shadow analysis, sky view factor Precision planning Shadow analysis Precinct structure plan height constraints 4D historic imagery Live travel information 3D cadaster. Planning application: Planning Support System based building envelope control, viewshed analysis, 360°visualization, integration with public health analysis, urban skyline current future.	Web services	Operational Proof of concept/Prototype

						Real time indoor positioning			
<u>New South (NSW)</u>	City	Buildings 3D Road Segments 3D (lines) Road Segments 2D (polygons) Railway 3D (lines) Hydroplanes 3D (lines) Terrain (grid) Trees 3D (point) Vegetation coverage 2D (polygon) Footpath 2D (lines) 4D model (3D model and time) Data from telco, water and energy utilities real time data	Buildings; Roads; Vegetation. 3D City Models	(LOD2, no semantics, surfaces)	3D analysis urban liveability and climate adaptability use cases.	Virtual environment (commonly Cesium) Shared data management delivery platform Data61 Data federation approach. Open sourced TerriaJS	Depends on the data sets between 1 and 4 years	Operational Under improvement (new features)	
<u>Digital Twin Victoria (DTV)</u>	City	an extensive catalogue of open data from across local, state and federal government, more than 4,000 datasets	Buildings; Roads; Vegetation		Visualize a DT model of Victoria. Collection of 3D spatial data.	Virtual environment (commonly Cesium) Shared data management delivery platform Data61 Data federation approach.	Visualizations of renewable energy capacity flooding data Scenario modelling for infrastructure development Real-time data for emergency management	Depends on the data sets between 1 and 4 years	Operational Under improvement (new features)

						Open sourced TerriaJS			
Liveable City Digital Twin (Diakite et al., 2022)	Liverpool precinct in Western Sydney	Buildings, terrain, transportation, water	Semantic 3D city model	LoD2	Create a City Digital Twin through data integration of different input geometrical and thematic data.	Cesium For visualizations (Manipulation and analysis QGIS) Connection between Cesium and database using API. Connection of PostgreSQL and Cesium with the help of python library Flask.	3D shadow analysis. Sun exposure and tree coverage. DT data update from the front- end. Canopy management using existing trees' shadows. Visualization of sensor feeds. Pedestrian flows.	Update of the DT using IoT data (real time interaction)	Operational and under development
NUS-FRS (Pradeep Alva et al., 2022)	District scale university campus	City Energy Analyst (CEA) Building data (footprint, height information from OSM) Real-time data Energy demand simulation results. Occupancy rates from WiFi data	3D campus model BIM model (Ifc2CityGML)	LoD1	Energy systems management on a district scale.	Digital Twin Dashboard based cesium web map application	post-pandemic and climate change scenarios		Operational
<u>EnSysLE project</u>	Germany counties: Dithmarschen,	2D and 3D geospatial data	CityGML 3D city models Buildings attributes		SDI to handle building energy based on a UDT	3D web application using Cesium	Energy simulations, heating and electric demands,		Operational, limited access to some data

(T. Santhanavanich et al., 2022)	Ilm-Kreis and Ludwigsburg.	application using (OGC) standards.	rooftop solar potential.
----------------------------------	----------------------------	------------------------------------	--------------------------

APPENDIX B

City2Twin Installation and reproducibility

This appendix describes how City2Twin was used throughout this work and how reproducibility was ensured across the different data integration scenarios developed in the thesis.

City2Twin is an open-source platform developed at the GeoScITY Lab and made available through an internal [ULiège GitLab repository](#). It supports both (i) standalone client-based workflows for visualization and interactive exploration and (ii) database deployments enabling advanced data integration and querying, including spatial filtering and attribute updates. The objective of this appendix is not to replace the official installation manuals. Instead, it documents the experimental environment, configuration choices, and usage patterns adopted in this thesis to facilitate both understanding and reproducibility. Detailed installation instructions and documentation are provided in README, together with a description of the implemented functionalities.

For transparency, the steps reported here focus on the elements required to use the platform, while implementation details remain referenced to the repository documentation.

B.1 Installation Overview

City2Twin can be deployed in two different configurations, depending on the user's need.

- Client only: This configuration is used for visualization, inspection, and manipulation of CityJSON files, GeoJSON, and IFC models. It allows users to load CityJSON files directly without requiring a database backend.
- Client and Backend configuration (with CJDB v2.1.0): For advanced querying and spatial filters, City2Twin can be connected to a backend relying on CJDB. In this case, CityJSON datasets must be imported in the [DB following instructions](#).

The step-by-step installation instructions for both configurations are provided in the [City2Twin GitLab repository](#).

B.2 City2Twin Usage for 3D City Models

City2Twin as a client supports CityJSON models and provides functionalities similar to the one presented to existing CityJSON viewers, while extending them with additional features, including

- Georeferencing at import time, when the CityJSON model does not define a CRS.
- Attribute based filtering, allowing users to explore subsets of city objects.
- Attribute classification, enabling the grouping or categorization of objects based on their attributes.

Spatial filtering and querying, allowing users to explore basics spatial filters (DWITHIN, DWITHOUT, INTERSECTS, DISJOINT). This functionality has not yet been integrated on the client's side. It must be executed on the server side of the database using PostgreSQL's spatial extension "PostGIS" and the "ground_geometry" column in CJDB, which contains the objects' footprints in PostGIS geometry format.

These functionalities were used for visualization, validation, and exploratory analysis of 3D city models throughout the thesis.

B.3 Reproducibility of the implemented data integration scenarios

A key contribution of this work is to ensure that the proposed data integration levels are reproducible across different contexts and adaptable to a variety of application settings.

For each data integration scenario developed in the thesis (corresponding to different levels of integration-based use case), a dedicated client template was developed. Each template specifies:

- The expected structure of the input data (e.g., JSON files for simulation outputs, and IFC for BIM models)
- The associated CityJSON and database schema conventions (i.e., mapping the simulation outputs to the CityJSON model, including the addition of new energy attributes, which can be stored in the DB on demand).
- The data processing and visualization workflow.

This approach ensures that each scenario can be reproduced, reused, or extended while avoiding inconsistencies caused by ad hoc configurations or lack of documentation.

For each template, test datasets are provided to allow users to validate the setup and reproduce the corresponding data integration scenario.

B.4 Database Optimization and preprocessing

In addition to the main City2Twin components, several scripts were developed to support DB optimization and geometry processing tasks required by the experiments.

These scripts include:

- Scripts for draping the 3D city model
- Scripts for optimizing CJDB geometry storage, particularly in cases where the *ground_geometry* is empty geometry.

B.5 Backend architecture considerations

The backend services implemented in this work are implemented using a Flask-based architecture. Flask provides a lightweight, flexible, and Python-native backend solution that is well suited to research-driven platforms, where modularity, reproducibility, and rapid experimentation are key requirements.

Each functional component (e.g., static 3D city models, IoT, user management) is implemented as a separate service. This modular organization supports scalability, maintenance, and reproducibility, and aligns with the scenario approach described earlier.

B.6 City2Twin Client Access

The City2Twin client is hosted on a university server and is accessible at the following [link](#). Users can upload their own datasets and directly benefit from the functionalities provided by the platform. Once the data are loaded, the 3D scene is automatically centered on the Outremeuse model. This model is stored in the CJDB database, enabling direct access to both attribute and spatial querying and visualization capabilities.

The City2Twin clients dedicated to energy analysis, BIM integration, and solar potential assessment can also be used locally, provided that the basic installation and execution instructions are followed.

APPENDIX C

Key Advantages and Limitations of UDT Integration Levels According to the Scenario Implementations

Table C.1. Key Advantages and limitations of the data integration levels for Scenario 1.

Levels	Key advantages	Key limitations
Level 1	- Ensure semantic interoperability → Enables carrying a carefully selected subset of IFC information into CityGML to support concrete use cases (e.g., energy analysis, routing, planning) without importing full BIM complexity. - A single ADE can serve multiple domains, reducing the need for bespoke schemas and improving consistency across projects. - Documented modelling choices → ADE design requires explicit, published decisions about semantics and geometry, improving transparency and comparability across implementations. - Standard alignment and version compatibility → Useful features can be aligned with future CityGML releases to maintain compatibility and foster interoperability.	- Data sparsity and inconsistency in IFC → Real-world IFC files often lack populated attributes or contain misclassifications, limiting automatic population of ADE fields. - No one-to-one mappings for many features → Some required attributes are not stored explicitly in IFC and require derivation or non-trivial processing, which can lead to divergent interpretations. - Limited tools → CityGML 3.0 and ADEs have limited software support, constraining semantic validation, editing, and adoption, and leading to ad-hoc solutions and workaround workflows. - Complexity and expertise → Designing and maintaining an ADE demands specialized expertise in both BIM and GIS modelling, increasing implementation overhead. - Data sparsity and misclassification → Real-world IFC files often have missing or misclassified semantics, limiting automatic population of ADE fields. - Validation gaps → Automated validation for CityGML 3.0 and ADEs are still in preliminary stage, limiting semantic correctness.
Level 2	- Harmonized, lightweight datasets suitable for visualization and analysis in 3D city models. - Maintaining key BIM semantics while ensuring schema compatibility. - Smooth integration into platforms like City2Twin that rely on CityJSON structures.	- Data Loss. - Semantic granularity can be reduced when IFC classes have no direct CityGML equivalent. - Dependence on preprocessing accurate georeferencement, CRS alignment, and model quality - Updates require re-conversion.

Level 3	- Preserves complete IFC semantics and property data for detailed analysis. - Provides interactive visualization and querying directly in a web browser. - No conversion steps are required avoiding thus the semantic loss - Scales to collaborative, real-time visualization scenarios.	- Rendering and memory performance may be affected when visualizing high-complexity IFC datasets in the browser. - Parsing time varies with the number of property sets and nested relationships. - Georeferencing accuracy still depends on correct IfcMapConversion information. - Less suited for urban and spatial analysis, no links between the two models.
----------------	--	--

Table C.2. Key Advantages and limitations of the data integration levels for Scenario 2.

Levels	Key advantages	Key limitations
Level 1	- Self-contained file-based model with embedded dynamic attributes. - Schema interoperability → Aligned with CityGML 3.0 Dynamizer module → Lightweight CityJSON extension. - Ease of visualization. - Standard compliance → CityJSON v2.0.	- File scalability. - Static updates. - Limited multi-user access. - Limited analytical depth.
Level 2	- CityGML 3.0 compliant representation. - Database-driven. - Hybrid dynamics → Embedded GenericTimeseries for offline use; use SensorConnection where single Datastreams exist. - Explicit linkage → Explicit attributeRef links between attributes and their time series. - Semantic Interoperability. - Straightforward migration → once each variable is available as an individual Datastream, you can switch from embedded data to live connections.	- Limited tool support few viewers or ETL tools interpret “Dynamizer”; many only show static attributes. - Constrained live integration. - Heavier files → Embedded time series increase size and require backfill/exports to update. - Synchronization challenges → Potential divergence between embedded values and the STA source without automated synchronization. - Strict CityGML schema conformance requirements → strict element ordering, namespaces, and correct attributeRef targets are required.
Level 3	- Real-time synchronization. - Scalability and modularity. - Standards-based interoperability. - Flexible analytics. - Lightweight 3D model.	- Dependence on network and server. - System complexity. - Data consistency challenges. - Partial standard alignment across tools.

Table C.3. Key pros and cons of the data integration levels for Scenario 3.

Levels	Key advantages	Key limitations
Level 1	- Standards compliance and semantic rigor→ Uses a formal Application Domain Extension (Energy ADE 2.0/Resources) within CityGML 2.0, ensuring schema validity, semantic interoperability, and stable identifiers that support reproducible analyses and cross-analysis. - Rich expressiveness and representation→ Support resource types, units, status, enabling transparent interpretation of energy metrics and their temporal structure at the building scale. - Persistent, queryable UDT backbone. - Conceptual extensibility. - Interoperable data exchange - Semantically consistent reference datasets.	- Limited native client support. - Loading and performance costs. - Heterogeneous service and infrastructure interoperability. - System requirements →Deployment entails database administration (PostgreSQL/3DCityDB), schema/version management, and alignment with the Energy ADE 2.0 tool ecosystem. - Non-incremental updates→ Update typically require re-ingestion and re-export via ETL. Scenario refreshes typically require re-running the ETL; continuous, API-driven updates are not intrinsic to Level 1. - Expertise requirements→ Correct ADE modelling, mapping, and validation demand specialist skills; misconfiguration risks schema non-compliance or semantic drift.
Level 2	- Semantic consistency and schema stability - Lightweight and standards-compliant - Reusable and scalable outputs - Automated, repeatable workflow - Seamless integration with the platform	- Static and pre-computed data. - Limited interactivity - Dependence on ETL framework - No direct link to the UDT client
Level 3-OGC STA standard-based	- System to system interoperability via OGC standardized infrastructure - Dynamic query and selective on-demand access. - Robust spatial and semantic linkage. - Extensibility → Additional data types (temporal series, multiple metrics) can be added.	- Higher and complex implementation. - Performance for requesting data. - Partial deployment of STA capabilities. - Complexity for non-expert users.
Level 3-Customized architecture	- Fully lightweight and client-side. - Flexible and user-driven visualization and analysis. - Multi-scale and responsive. - Avoids heavy dependencies.	- Client-side memory and performance constraints. - No persistent data storage. - Limited semantic interoperability.

BIBLIOGRAPHY

- Abdalla, R., 2016. Geospatial Data Integration, in: Abdalla, R. (Ed.), *Introduction to Geospatial Information and Communication Technology (GeoICT)*. Springer International Publishing, Cham, pp. 105–124. https://doi.org/10.1007/978-3-319-33603-9_6
- Abdelrahman, M., Macatulad, E., Lei, B., Quintana, M., Miller, C., Biljecki, F., 2025. What is a Digital Twin anyway? Deriving the definition for the built environment from over 15,000 scientific publications. *Building and Environment* 274, 112748. <https://doi.org/10.1016/j.buildenv.2025.112748>
- Acharya, S., Khan, A.A., Päiväranta, T., 2024. Interoperability levels and challenges of digital twins in cyber–physical systems. *Journal of Industrial Information Integration* 42, 100714. <https://doi.org/10.1016/j.jii.2024.100714>
- Adreani, L., Colombo, C., Fanfani, M., Nesi, P., Pantaleo, G., Pisanu, R., 2022. A Photorealistic 3D City Modeling Framework for Smart City Digital Twin, in: *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*. Presented at the 2022 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 299–304. <https://doi.org/10.1109/SMARTCOMP55677.2022.00071>
- Agostinelli, S., Cumo, F., Guidi, G., Tomazzoli, C., 2021. Cyber-Physical Systems Improving Building Energy Management: Digital Twin and Artificial Intelligence. *Energies* 14, 2338. <https://doi.org/10.3390/en14082338>
- Agugiaro, G., Benner, J., Cipriano, P., Nouvel, R., 2018. The Energy Application Domain Extension for CityGML: enhancing interoperability for urban energy simulations. *Open geospatial data, softw. stand.* 3, 2. <https://doi.org/10.1186/s40965-018-0042-y>
- Agugiaro, G., Padsala, R., 2025. A proposal to update and enhance the CityGML Energy Application Domain Extension. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* X-4-W6-2025, 1–8. <https://doi.org/10.5194/isprs-annals-X-4-W6-2025-1-2025>
- Alva, P., Biljecki, F., Stouffs, R., 2022. USE CASES FOR DISTRICT-SCALE URBAN DIGITAL TWINS. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLVIII-4/W4-2022, 5–12. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W4-2022-5-2022>
- Alva, Pradeep, Mosteiro Romero, M., Miller, C., Stouffs, R., 2022. Digital Twin-Based Resilience Evaluation of District-Scale Archetypes. pp. 525–534. <https://doi.org/10.52842/conf.caadria.2022.1.525>
- Ammar, A., Nassereddine, H., AbdulBaky, N., AbouKansour, A., Tannoury, J., Urban, H., Schranz, C., 2022. Digital Twins in the Construction Industry: A Perspective of Practitioners and Building Authority. *Front. Built Environ.* 8, 834671. <https://doi.org/10.3389/fbuil.2022.834671>
- Atkinson, R.A., Zaborowski, P., Noardo, F., Simonis, I., 2022. SMART CITIES – SYSTEMS OF SYSTEMS INTEROPERABILITY AND OGC ENABLERS. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* X-4/W3-2022, 19–26. <https://doi.org/10.5194/isprs-annals-X-4-W3-2022-19-2022>
- Ávila Eça de Matos, B., Dane, G.Z., Van Tilburg, T., Verstappen, J., de Vries, B., 2022. State-of-the-Art of the Urban Digital Twin Ecosystem in the Netherlands: 3rd International Smart Cities in Smart Regions Conference. *Smart Cities in Smart Regions 2022 Conference Proceedings*.
- Bacher, U., 2022. Hybrid Aerial Sensor Data as Basis for a Geospatial Digital Twin. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information*

- Sciences 43B4, 653–659. <https://doi.org/10.5194/isprs-archives-XLIII-B4-2022-653-2022>
- Ballouch, Z., Jeddoub, I., Hajji, R., Kasprzyk, J.-P., Billen, R., 2024. Towards a Digital Twin of Liege: The Core 3D Model based on Semantic Segmentation and Automated Modeling of LiDAR Point Clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* X-4-W4-2024, 13–20. <https://doi.org/10.5194/isprs-annals-X-4-W4-2024-13-2024>
- Batty, M., 2018. Digital twins. *Environment and Planning B: Urban Analytics and City Science* 45, 817–820. <https://doi.org/10.1177/2399808318796416>
- Batty, M., Hudson-Smith, A., 2006. Digital Cornucopias: Changing Conceptions of the Virtual City. *Environ Plann B Plann Des* 33, 799–802. <https://doi.org/10.1068/b3306ed>
- Bauer, M., Cirillo, F., Fürst, J., Solmaz, G., Kovacs, E., 2021. Urban Digital Twins – A FIWARE-based model. *at - Automatisierungstechnik* 69, 1106–1115. <https://doi.org/10.1515/auto-2021-0083>
- Biljecki, F., Lim, J., Crawford, J., Moraru, D., Tauscher, H., Konde, A., Adouane, K., Lawrence, S., Janssen, P., Stouffs, R., 2021. Extending CityGML for IFC-sourced 3D city models. *Automation in Construction* 121, 103440. <https://doi.org/10.1016/j.autcon.2020.103440>
- Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015. Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information* 4, 2842–2889. <https://doi.org/10.3390/ijgi4042842>
- Billen, R., Cutting-Decelle, A., Marina, O., Almeida, J.-P., Caglioni, M., Falquet, G., Leduc, T., Métral, C., Moreau, G., Perret, J., Rabino, G., García, R., Yatskiv, I., Zlatanova, S., 2014. 3D City Models and urban information: Current issues and perspectives. <https://doi.org/10.1051/TU0801/201400001>
- Boje, C., Guerriero, A., Kubicki, S., Rezgüi, Y., 2020. Towards a semantic Construction Digital Twin: Directions for future research. *Automation in Construction* 114, 103179. <https://doi.org/10.1016/j.autcon.2020.103179>
- Bolton, A., Butler, L., Dabson, I., Enzer, M., Evans, M., Fenemore, T., Harradence, F., Keaney, E., Kemp, A., Luck, A., Pawsey, N., Saville, S., Schooling, J., Sharp, M., Smith, T., Tennison, J., Whyte, J., Wilson, A., Makri, C., 2018. Gemini Principles (Report). CDBB. <https://doi.org/10.17863/CAM.32260>
- Botín-Sanabria, D.M., Mihaita, A.-S., Peimbert-García, R.E., Ramírez-Moreno, M.A., Ramírez-Mendoza, R.A., Lozoya-Santos, J. de J., 2022. Digital Twin Technology Challenges and Applications: A Comprehensive Review. *Remote Sensing* 14, 1335. <https://doi.org/10.3390/rs14061335>
- Boumhidi, K., Nys, G.-A., Hajji, R., 2024. Integrating Dynamic Data with 3D City Models via CityJSON Extension, in: Kolbe, T.H., Donaubauer, A., Beil, C. (Eds.), *Recent Advances in 3D Geoinformation Science*. Springer Nature Switzerland, Cham, pp. 745–758. https://doi.org/10.1007/978-3-031-43699-4_45
- Caprari, G., 2022. Digital Twin for Urban Planning in the Green Deal Era: A State of the Art and Future Perspectives. *Sustainability* 14, 6263. <https://doi.org/10.3390/su14106263>
- Chadzynski, A., Krdzavac, N., Farazi, F., Lim, M.Q., Li, S., Grisiute, A., Herthogs, P., von Richthofen, A., Cairns, S., Kraft, M., 2021. Semantic 3D City Database — An enabler for a dynamic geospatial knowledge graph. *Energy and AI* 6, 100106. <https://doi.org/10.1016/j.egyai.2021.100106>
- Channi, H.K., Kumar, P., Safdarian, F., 2025. Chapter 10 - Planning and building digital twins for smart cities, in: Iyer, S., Nayyar, A., Paul, A., Naved, M. (Eds.), *Digital Twins for Smart Cities and Villages*. Elsevier, pp. 205–225. <https://doi.org/10.1016/B978-0-443-28884-5.00010-5>

- Charitonidou, M., 2022. Urban scale digital twins in data-driven society: Challenging digital universalism in urban planning decision-making. *International Journal of Architectural Computing* 20, 238–253. <https://doi.org/10.1177/14780771211070005>
- Chaturvedi, K., 2021. Integration and Management of Time-dependent Properties with Semantic 3D City Models.
- Chaturvedi, Kanishk, Matheus, A., Nguyen, S.H., Kolbe, T.H., 2019. Securing Spatial Data Infrastructures for Distributed Smart City applications and services. *Future Generation Computer Systems* 101, 723–736. <https://doi.org/10.1016/j.future.2019.07.002>
- Chaturvedi, K., Willenborg, B., Sindram, M., Kolbe, T.H., 2017. SOLAR POTENTIAL ANALYSIS AND INTEGRATION OF THE TIME-DEPENDENT SIMULATION RESULTS FOR SEMANTIC 3D CITY MODELS USING DYNAMIZERS. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* IV-4/W5, 25–32. <https://doi.org/10.5194/isprs-annals-IV-4-W5-25-2017>
- Chaturvedi, K., Yao, Z., Kolbe, T.H., 2019. INTEGRATED MANAGEMENT AND VISUALIZATION OF STATIC AND DYNAMIC PROPERTIES OF SEMANTIC 3D CITY MODELS. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLII-4/W17, 7–14. <https://doi.org/10.5194/isprs-archives-XLII-4-W17-7-2019>
- Chatzinikolaou, E., Pispidikis, I., Dimopoulou, E., 2020. A SEMANTICALLY ENRICHED AND WEB-BASED 3D ENERGY MODEL VISUALIZATION AND RETRIEVAL FOR SMART BUILDING IMPLEMENTATION USING CITYGML AND DYNAMIZER ADE, in: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Presented at the ISPRS TC IV
3rd BIM/GIS Integration Workshop and 15th 3D GeoInfo Conference 2020 (Volume VI-4/W1-2020) - 7–11 September 2020, London, UK, Copernicus GmbH, pp. 53–60. <https://doi.org/10.5194/isprs-annals-VI-4-W1-2020-53-2020>
- Chehrehbargh, F.J., 2022. Fishermans Bend Digital Twin: Centre for Spatial Data Infrastructures and Land Administration, The University of Melbourne [WWW Document]. CSDILA. URL <https://eng.unimelb.edu.au/csdila/projects/digital-twin/fishermans-bend> (accessed 2.6.23).
- Chen, R., 2011. The development of 3D city model and its applications in urban planning, in: 2011 19th International Conference on Geoinformatics. Presented at the 2011 19th International Conference on Geoinformatics, pp. 1–5. <https://doi.org/10.1109/GeoInformatics.2011.5981007>
- Cirillo, F., Solmaz, G., Berz, E.L., Bauer, M., Cheng, B., Kovacs, E., 2019. A Standard-Based Open Source IoT Platform: FIWARE. *IEEE Internet of Things Magazine* 2, 12–18. <https://doi.org/10.1109/IOTM.0001.1800022>
- City information modelling and urban digital twins | IEC [WWW Document], n.d. URL <https://www.iec.ch/basecamp/city-information-modelling-and-urban-digital-twins> (accessed 11.30.22).
- Corongiu, M., Tucci, G., Santoro, E., Kourounioti, O., 2018. Data integration of different domains in geo-information management: A railway infrastructure case study. Presented at the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, pp. 193–200. <https://doi.org/10.5194/isprs-archives-XLII-4-121-2018>
- Cureton, P., Dunn, N., 2021. Chapter 14 - Digital twins of cities and evasive futures, in: Aurigi, A., Odendaal, N. (Eds.), *Shaping Smart for Better Cities*. Academic Press, pp. 267–282. <https://doi.org/10.1016/B978-0-12-818636-7.00017-2>
- Dembski, F., Wössner, U., Letzgus, M., Ruddat, M., Yamu, C., 2020. Urban Digital Twins for Smart Cities and Citizens: The Case Study of Herrenberg, Germany. *Sustainability* 12, 2307. <https://doi.org/10.3390/su12062307>

- Deng, T., Zhang, K., Shen, Z.-J. (Max), 2021. A systematic review of a digital twin city: A new pattern of urban governance toward smart cities. *Journal of Management Science and Engineering* 6, 125–134. <https://doi.org/10.1016/j.jmse.2021.03.003>
- Deng, Z., Hijazi, I., Kolbe, T., 2025. CityGML 3.0 as a Hub: Integrating BIM, GIS, and Point Cloud Data for 3D Streetspace Modeling Comprising Roads, Bridges and Tunnels. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* X-4/W6-2025, 65–72. <https://doi.org/10.5194/isprs-annals-X-4-W6-2025-65-2025>
- Deprêtre, A., Jacquinod, F., Mielniczek, A., 2022. EXPLORING DIGITAL TWIN ADAPTATION TO THE URBAN ENVIRONMENT: COMPARISON WITH CIM TO AVOID SILO-BASED APPROACHES. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* V-4-2022, 337–344. <https://doi.org/10.5194/isprs-annals-V-4-2022-337-2022>
- Deren, L., Wenbo, Y., Zhenfeng, S., 2021. Smart city based on digital twins. *Comput.Urban Sci.* 1, 4. <https://doi.org/10.1007/s43762-021-00005-y>
- Diakite, A.A., Ng, L., Barton, J., Rigby, M., Williams, K., Barr, S., Zlatanova, S., 2022. LIVEABLE CITY DIGITAL TWIN: A PILOT PROJECT FOR THE CITY OF LIVERPOOL (NSW, AUSTRALIA). *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* X-4/W2-2022, 45–52. <https://doi.org/10.5194/isprs-annals-X-4-W2-2022-45-2022>
- Dimitrov, H., Petrova-Antonova, D., 2021. 3D city model as a first step towards digital twin of Sofia City. Presented at the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, pp. 23–30. <https://doi.org/10.5194/isprs-archives-XLIII-B4-2021-23-2021>
- Doi, H., Yarroudh, A., Jeddoub, I., Hajji, R., Billen, R., 2024. Automatic Detection and 3D Modeling of City Furniture Objects using LiDAR and Imagery Mobile Mapping Data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLVIII-2/W8-2024, 125–132. <https://doi.org/10.5194/isprs-archives-XLVIII-2-W8-2024-125-2024>
- Döllner, J., 2020. Geospatial Artificial Intelligence: Potentials of Machine Learning for 3D Point Clouds and Geospatial Digital Twins. *PFG – Journal of Photogrammetry Remote Sensing and Geoinformation Science* 88. <https://doi.org/10.1007/s41064-020-00102-3>
- Döllner, J., Buchholz, H., 2005. Continuous level-of-detail modeling of buildings in 3D city models, in: *Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems, GIS '05*. Association for Computing Machinery, New York, NY, USA, pp. 173–181. <https://doi.org/10.1145/1097064.1097089>
- Döllner, J., Kolbe, T.H., Liecke, F., Sgouros, T., Teichmann, K., n.d. *The Virtual 3D City Model of Berlin* -.
- Ellul, C., Stoter, J., Bucher, B., 2022. LOCATION-ENABLED DIGITAL TWINS – UNDERSTANDING THE ROLE OF NMCAS IN A EUROPEAN CONTEXT. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* X-4/W2-2022, 53–60. <https://doi.org/10.5194/isprs-annals-X-4-W2-2022-53-2022>
- F. Biljecki, 2017. Level of detail in 3D city models. Delft University of Technology. <https://doi.org/10.4233/UUID:F12931B7-5113-47EF-BFD4-688AAE3BE248>
- Ferré-Bigorra, J., Casals, M., Gangoles, M., 2022. The adoption of urban digital twins. *Cities* 131, 103905. <https://doi.org/10.1016/j.cities.2022.103905>
- Fricke, A., Döllner, J., Asche, H., Isaac, F., 2023. Optimising a Virtual 3D City Model for Citizen-Based Urban Development in East Jerusalem, in: Gervasi, O., Murgante, B., Rocha, A.M.A.C., Garau, C., Scorza, F., Karaca, Y., Torre, C.M. (Eds.), *Computational Science and Its Applications – ICCSA 2023 Workshops, Lecture Notes in Computer*

- Science. Springer Nature Switzerland, Cham, pp. 521–535. https://doi.org/10.1007/978-3-031-37114-1_36
- Fuller, A., Fan, Z., Day, C., Barlow, C., 2020. Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Access* 8, 108952–108971. <https://doi.org/10.1109/ACCESS.2020.2998358>
- Gil, J., Petrova-Antonova, D., Kemp, G.J., 2024. Redefining urban digital twins for the federated data spaces ecosystem: A perspective. *Environment and Planning B* 23998083241302578. <https://doi.org/10.1177/23998083241302578>
- Gitahi, J., Kolbe, T.H., 2024. Requirements for Web-Based 4D Visualisation of Integrated 3D City Models and Sensor Data in Urban Digital Twins, in: Kolbe, T.H., Donaubauer, A., Beil, C. (Eds.), *Recent Advances in 3D Geoinformation Science, Lecture Notes in Geoinformation and Cartography*. Springer Nature Switzerland, Cham, pp. 707–725. https://doi.org/10.1007/978-3-031-43699-4_43
- Grieves, M., 2022. Intelligent digital twins and the development and management of complex systems. *digitaltwin* 2, 8. <https://doi.org/10.12688/digitaltwin.17574.1>
- Grieves, M., 2016. Origins of the Digital Twin Concept. <https://doi.org/10.13140/RG.2.2.26367.61609>
- Hagedorn, J.D. & B., 2007. Integrating urban GIS, CAD, and BIM data by service-based virtual 3D city models, in: *Urban and Regional Data Management*. CRC Press.
- Hakim, A., Ohori, K., Vaart, J., El Yamani, S., Stoter, J., 2024. Enhancing Georeferencing of IFC Models through Surveyed Points Integration. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLVIII-4/W11-2024*, 41–47. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W11-2024-41-2024>
- Hämäläinen, M., 2021. Urban development with dynamic digital twins in Helsinki city. *IET Smart Cities* 3, 201–210. <https://doi.org/10.1049/smc2.12015>
- Hämäläinen, M., 2020. Smart city development with digital twin technology. <https://doi.org/10.18690/978-961-286-362-3.20>
- Haraguchi, M., Funahashi, T., Biljecki, F., 2024. Assessing governance implications of city digital twin technology: A maturity model approach. *Technological Forecasting and Social Change* 204, 123409. <https://doi.org/10.1016/j.techfore.2024.123409>
- Herbert, G., Chen, X., 2015. A comparison of usefulness of 2D and 3D representations of urban planning. *Cartography and Geographic Information Science* 42, 22–32. <https://doi.org/10.1080/15230406.2014.987694>
- Herzog, R., Degkwitz, T., Verma, T., 2025. The Urban Model Platform: A Public Backbone for Modeling and Simulation in Urban Digital Twins. <https://doi.org/10.48550/arXiv.2506.10964>
- Hijazi, I., Donaubauer, A., Hamm, A., Falkenstein, A., Kolbe, T., 2022. URBAN GROWTH SIMULATION USING URBAN DYNAMICS AND CITYGML: A USE CASE FROM THE CITY OF MUNICH. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences X-4/W2-2022*, 97–104. <https://doi.org/10.5194/isprs-annals-X-4-W2-2022-97-2022>
- Hristov, P.O., Petrova-Antonova, D., Ilieva, S., Rizov, R., 2022. ENABLING CITY DIGITAL TWINS THROUGH URBAN LIVING LABS. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLIII-B1-2022*, 151–156. <https://doi.org/10.5194/isprs-archives-XLIII-B1-2022-151-2022>
- Hu, Y., 2017. Spatial Data Infrastructures. *Geographic Information Science & Technology Body of Knowledge 2017*. <https://doi.org/10.22224/gistbok/2017.2.1>
- Ivanov, S., Nikolskaya, K., Radchenko, G., Sokolinsky, L., Zymbler, M., 2020. Digital Twin of City: Concept Overview, in: *2020 Global Smart Industry Conference (GloSIC)*.

- Presented at the 2020 Global Smart Industry Conference (GloSIC), pp. 178–186.
<https://doi.org/10.1109/GloSIC50886.2020.9267879>
- Janowicz, K., Schade, S., Bröring, A., Keßler, C., Maué, P., Stasch, C., 2010. Semantic Enablement for Spatial Data Infrastructures. *Transactions in GIS* 14, 111–129.
<https://doi.org/10.1111/j.1467-9671.2010.01186.x>
- Jeddoub, I., Nys, G.-A., Hajji, R., Billen, R., 2024. Data integration across urban digital twin lifecycle: a comprehensive review of current initiatives. *Annals of GIS* 0, 1–20.
<https://doi.org/10.1080/19475683.2024.2416135>
- Jeddoub, I., Nys, G.-A., Hajji, R., Billen, R., 2023. Digital Twins for cities: Analyzing the gap between concepts and current implementations with a specific focus on data integration. *International Journal of Applied Earth Observation and Geoinformation* 122, 103440.
<https://doi.org/10.1016/j.jag.2023.103440>
- Jeddoub, I., Roquet, M., Oudadda, Y., Hajji, R., Dewallef, P., Billen, R., 2025. A Lightweight Framework for Seamless Integration of Building Energy Simulations into Urban Digital Twins. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLVIII-4-W15-2025*, 77–84. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W15-2025-77-2025>
- Kanna, K., Kolbe, T., 2025. Automatic Enrichment of Semantic 3D City Models using Large Language Models. <https://doi.org/10.5194/isprs-annals-X-4-W6-2025-105-2025>
- Kasprzyk, J.-P., Nys, G.-A., Billen, R., 2024. Towards a multi-database CityGML environment adapted to big geodata issues of urban digital twins. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLVIII-4-W10-2024*, 101–106. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W10-2024-101-2024>
- Katal, A., Mortezaazadeh, M., Wang, L. (Leon), Yu, H., 2022. Urban building energy and microclimate modeling – From 3D city generation to dynamic simulations. *Energy* 251, 123817. <https://doi.org/10.1016/j.energy.2022.123817>
- Ketzler, B., Naserentin, V., Latino, F., Zangelidis, C., Thuvander, L., Logg, A., 2020. Digital Twins for Cities: A State of the Art Review. *Built Environment* 46, 547–573.
<https://doi.org/10.2148/benv.46.4.547>
- Khawte, S.S., Koeva, M.N., Gevaert, C.M., Oude Elberink, S., Pedro, A.A., 2022. DIGITAL TWIN CREATION FOR SLUMS IN BRAZIL BASED ON UAV DATA. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLVIII-4/W4-2022*, 75–81.
<https://doi.org/10.5194/isprs-archives-XLVIII-4-W4-2022-75-2022>
- Knezevic, M., Donaubaue, A., Moshrefzadeh, M., Kolbe, T.H., 2022. MANAGING URBAN DIGITAL TWINS WITH AN EXTENDED CATALOG SERVICE. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. X-4/W3-2022*, 119–126.
<https://doi.org/10.5194/isprs-annals-X-4-W3-2022-119-2022>
- Kogut, P., Heijden, R., 2025. Digital Twins for Urban Governance: General Desires, Expectations, Challenges. pp. 9–31. https://doi.org/10.1007/978-3-031-81451-8_2
- Kolbe, D.T.H., Moshrefzadeh, M., Chaturvedi, K., Donaubaue, D.A., n.d. Updated Version, August 2020.
- Kolbe, T.H., Donaubaue, A., 2021. Semantic 3D City Modeling and BIM, in: Shi, W., Goodchild, M.F., Batty, M., Kwan, M.-P., Zhang, A. (Eds.), *Urban Informatics, The Urban Book Series*. Springer, Singapore, pp. 609–636. https://doi.org/10.1007/978-981-15-8983-6_34
- Kotsev, A., Minghini, M., Tomas, R., Cetl, V., Lutz, M., 2020. From Spatial Data Infrastructures to Data Spaces—A Technological Perspective on the Evolution of European SDIs. *ISPRS International Journal of Geo-Information* 9, 176.
<https://doi.org/10.3390/ijgi9030176>

- Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W., 2018. Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine, 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018 51, 1016–1022. <https://doi.org/10.1016/j.ifacol.2018.08.474>
- Kumar, K., Ledoux, H., Commandeur, T., Stoter, J., 2017. MODELLING URBAN NOISE IN CITYGML ADE: CASE OF THE NETHERLANDS. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences IV-4/W5, 73–81. <https://doi.org/10.5194/isprs-annals-IV-4-W5-73-2017>
- Kutzner, T., Chaturvedi, K., Kolbe, T.H., 2020. CityGML 3.0: New Functions Open Up New Applications. PFG 88, 43–61. <https://doi.org/10.1007/s41064-020-00095-z>
- La Guardia, M., Koeva, M., 2023. Towards Digital Twinning on the Web: Heterogeneous 3D Data Fusion Based on Open-Source Structure. Remote Sensing 15, 721. <https://doi.org/10.3390/rs15030721>
- La Guardia, M., Koeva, M., D'Ippolito, F., Karam, S., 2022. 3D DATA INTEGRATION FOR WEB BASED OPEN SOURCE WebGL INTERACTIVE VISUALISATION. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLVIII-4-W4-2022, 89–94. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W4-2022-89-2022>
- Ledoux, H., Arroyo Ohori, K., Kumar, K., Dukai, B., Labetski, A., Vitalis, S., 2019. CityJSON: a compact and easy-to-use encoding of the CityGML data model. Open Geospatial Data, Software and Standards 4, 4. <https://doi.org/10.1186/s40965-019-0064-0>
- Ledoux, H., Biljecki, F., Dukai, B., Kumar, K., Peters, R., Stoter, J., Commandeur, T., 2021. 3dfier: automatic reconstruction of 3D city models. <https://doi.org/10.21105/joss.02866>
- Ledoux, H., Stavropoulou, G., Dukai, B., 2024. Streaming CityJSON datasets. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLVIII-4-W11-2024, 57–63. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W11-2024-57-2024>
- Lee, A., Lee, K.-W., Kim, K.-H., Shin, S.-W., 2022. A Geospatial Platform to Manage Large-Scale Individual Mobility for an Urban Digital Twin Platform. Remote Sensing 14, 723. <https://doi.org/10.3390/rs14030723>
- Lehner, H., Dorffner, L., 2020. Digital geoTwin Vienna: Towards a Digital Twin City as Geodata Hub. PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science 88, 63.
- Lehner, H., Kordasch, S.L., Glatz, C., Agugiaro, G., 2024. Digital geoTwin: A CityGML-Based Data Model for the Virtual Replica of the City of Vienna, in: Kolbe, T.H., Donaubauer, A., Beil, C. (Eds.), Recent Advances in 3D Geoinformation Science, Lecture Notes in Geoinformation and Cartography. Springer Nature Switzerland, Cham, pp. 517–541. https://doi.org/10.1007/978-3-031-43699-4_32
- Lehtola, V.V., Koeva, M., Elberink, S.O., Raposo, P., Virtanen, J.-P., Vahdatikhaki, F., Borsci, S., 2022. Digital twin of a city: Review of technology serving city needs. International Journal of Applied Earth Observation and Geoinformation 102915. <https://doi.org/10.1016/j.jag.2022.102915>
- Lei, B., Janssen, P., Stoter, J., Biljecki, F., 2023. Challenges of urban digital twins: A systematic review and a Delphi expert survey. Automation in Construction 147, 104716. <https://doi.org/10.1016/j.autcon.2022.104716>
- Lei, B., Stouffs, R., Biljecki, F., 2022. Assessing and benchmarking 3D city models. International Journal of Geographical Information Science. <https://doi.org/10.1080/13658816.2022.2140808>
- Lei, B., Su, Y., Biljecki, F., 2024. Humans As Sensors in Urban Digital Twins. pp. 693–706. https://doi.org/10.1007/978-3-031-43699-4_42

- Lei, B., Su, Y., Biljecki, F., n.d. Humans As Sensors in Urban Digital Twins.
- Li, W., Zlatanova, S., Diakite, A.A., Aleksandrov, M., Yan, J., 2020. Towards Integrating Heterogeneous Data: A Spatial DBMS Solution from a CRC-LCL Project in Australia. *ISPRS International Journal of Geo-Information* 9, 63. <https://doi.org/10.3390/ijgi9020063>
- Li, W., Zlatanova, S., Yan, J.J., Diakite, A., Aleksandrov, M., 2019. A GEO-DATABASE SOLUTION FOR THE MANAGEMENT AND ANALYSIS OF BUILDING MODEL WITH MULTI-SOURCE DATA FUSION. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-4-W20*, 55–63. <https://doi.org/10.5194/isprs-archives-XLII-4-W20-55-2019>
- Liu, X., Antwi-Afari, M., Li, J., Zhang, Y., Manu, P., 2025. BIM, IoT, and GIS integration in construction resource monitoring. *Automation in Construction* 174, 106149. <https://doi.org/10.1016/j.autcon.2025.106149>
- Lu, Q., Parlikad, A.K., Woodall, P., Don Ranasinghe, G., Xie, X., Liang, Z., Konstantinou, E., Heaton, J., Schooling, J., 2020. Developing a Digital Twin at Building and City Levels: Case Study of West Cambridge Campus. *Journal of Management in Engineering* 36, 05020004. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000763](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000763)
- Lu, Q., Parlikad, A.K., Woodall, P., Xie, X., Liang, Z., Konstantinou, E., Heaton, J., Schooling, J., 2019. Developing a dynamic digital twin at building and city levels: A case study of the West Cambridge campus. *Journal of Management in Engineering* 36. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000763](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000763)
- Macq, H., Jeddoub, I., Billen, R., 2025. Beyond Twinness and Twinning: Reflections on Urban Digital Twinning from the Namur 3D Project. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLVIII-4/W16-2025*, 51–58. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W16-2025-51-2025>
- Mohammadi, N., Taylor, J.E., 2017. Smart city digital twins, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI). Presented at the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–5. <https://doi.org/10.1109/SSCI.2017.8285439>
- Mylonas, G., Kalogeras, A.P., Kalogeras, G., Anagnostopoulos, C., Alexakos, C., Munoz, L., 2021. Digital Twins From Smart Manufacturing to Smart Cities: A Survey. *IEEE Access* PP, 1–1. <https://doi.org/10.1109/ACCESS.2021.3120843>
- Nguyen, S.H., Kolbe, T.H., 2022. PATH-TRACING SEMANTIC NETWORKS TO INTERPRET CHANGES IN SEMANTIC 3D CITY MODELS. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* X-4/W2-2022, 217–224. <https://doi.org/10.5194/isprs-annals-X-4-W2-2022-217-2022>
- Noardo, F., 2022. Multisource spatial data integration for use cases applications. *Transactions in GIS* 26, 2874–2913. <https://doi.org/10.1111/tgis.12987>
- Noardo, F., Arroyo Otori, K., Biljecki, F., Ellul, C., Harrie, L., Krijnen, T., Eriksson, H., van Liempt, J., Pla, M., Ruiz, A., Hintz, D., Krueger, N., Leoni, C., Leoz, L., Moraru, D., Vitalis, S., Willkomm, P., Stoter, J., 2021. Reference study of CityGML software support: The GeoBIM benchmark 2019—Part II. *Transactions in GIS* 25, 842–868. <https://doi.org/10.1111/tgis.12710>
- Noardo, F., Arroyo Otori, K., Biljecki, F., Ellul, C., Harrie, L., Krijnen, T., Kokla, M., Stoter, J., 2020. THE ISPRS-EUROSDR GEOBIM BENCHMARK 2019. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLIII-B5-2020, 227–233. <https://doi.org/10.5194/isprs-archives-XLIII-B5-2020-227-2020>
- Noardo, Francesca, Harrie, L., Arroyo Otori, K., Biljecki, F., Ellul, C., Krijnen, T., Eriksson, H., Guler, D., Hintz, D., Jadidi, M.A., Pla, M., Sanchez, S., Soini, V.-P., Stouffs, R., Tekavec, J., Stoter, J., 2020. Tools for BIM-GIS Integration (IFC Georeferencing and

- Conversions): Results from the GeoBIM Benchmark 2019. *IJGI* 9, 502. <https://doi.org/10.3390/ijgi9090502>
- Nochta, T., Parlikad, A., Schooling, J., Badstuber, N., Wahby, N., 2019. The local governance of digital technology – Implications for the city-scale digital twin (Report). CDBB. <https://doi.org/10.17863/CAM.43321>
- Nochta, T., Wan, L., Schooling, J.M., Parlikad, A.K., 2021. A Socio-Technical Perspective on Urban Analytics: The Case of City-Scale Digital Twins. *Journal of Urban Technology* 28, 263–287. <https://doi.org/10.1080/10630732.2020.1798177>
- Nys, G.-A., Billen, R., 2021. From consistency to flexibility: A simplified database schema for the management of CityJSON 3D city models. *Transactions in GIS* 25, 3048–3066. <https://doi.org/10.1111/tgis.12807>
- Omran, H., Ghaffarianhoseini, Amirhosein, Ghaffarianhoseini, Ali, Clements-Croome, D.J., 2022. The uptake of City Information Modelling (CIM): a comprehensive review of current implementations, challenges and future outlook. *Smart and Sustainable Built Environment ahead-of-print*. <https://doi.org/10.1108/SASBE-06-2022-0116>
- Padsala, R., Banerjee, S., Belayneh, T., Santhanavanich, T., Coors, V., 2023. Conceptualizing I3S encoding for OGC CityGML 3.0 Building Model. <https://doi.org/10.1145/3611314.3615920>
- Padsala, R., Falay, B., Hainoun, A., Coors, V., 2025. A Data-Driven Urban Digital Twin Approach for Evaluating Positive Energy District Potential Using OGC Standards in Stuttgart. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLVIII-4/W16-2025*, 67–74. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W16-2025-67-2025>
- Padsala, R., Gebetsroither-Geringer, E., Peters-Anders, J., Coors, V., 2021. INCEPTION of HARMONISING DATA SILOS and URBAN SIMULATION TOOLS USING 3D CITY MODELS for SUSTAINABLE MANAGEMENT of the URBAN FOOD WATER and ENERGY RESOURCES. Presented at the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 81–88. <https://doi.org/10.5194/isprs-annals-VIII-4-W1-2021-81-2021>
- Papyshev, G., Yarime, M., 2021. Exploring city digital twins as policy tools: A task-based approach to generating synthetic data on urban mobility. *Data & Policy* 3. <https://doi.org/10.1017/dap.2021.17>
- Petrova-Antonova, D., Ilieva, S., 2021. Digital Twin Modeling of Smart Cities, in: Ahram, T., Taiar, R., Langlois, K., Choplin, A. (Eds.), *Human Interaction, Emerging Technologies and Future Applications III, Advances in Intelligent Systems and Computing*. Springer International Publishing, Cham, pp. 384–390. https://doi.org/10.1007/978-3-030-55307-4_58
- Piras, G., Agostinelli, S., Muzi, F., 2024. Digital Twin Framework for Built Environment: A Review of Key Enablers. *Energies* 17, 436. <https://doi.org/10.3390/en17020436>
- Powalka, L., Poon, C., Xia, Y., Meines, S., Yan, L., Cai, Y., Stavropoulou, G., Dukai, B., Ledoux, H., 2023. cjdb: a simple, fast, and lean database solution for the CityGML data model.
- Raes, L., McAleer, S.R., 2025. Introduction: Defining Local Digital Twins, in: Raes, L., Ruston McAleer, S., Croket, I., Kogut, P., Brynskov, M., Lefever, S. (Eds.), *Decide Better: Open and Interoperable Local Digital Twins*. Springer Nature Switzerland, Cham, pp. 1–5. https://doi.org/10.1007/978-3-031-81451-8_1
- Raes, L., Michiels, P., Adolphi, T., Tampere, C., Dalianis, T., McAleer, S., Kogut, P., 2021. DUET: A Framework for Building Interoperable and Trusted Digital Twins of Smart Cities. *IEEE Internet Computing* 1–1. <https://doi.org/10.1109/MIC.2021.3060962>

- Rossknecht, M., Airaksinen, E., 2020. Concept and Evaluation of Heating Demand Prediction Based on 3D City Models and the CityGML Energy ADE—Case Study Helsinki. *ISPRS International Journal of Geo-Information* 9, 602. <https://doi.org/10.3390/ijgi9100602>
- Saeed, Z.O., Mancini, F., Glusac, T., Izadpanahi, P., 2022. Future City, Digital Twinning and the Urban Realm: A Systematic Literature Review. *Buildings* 12, 685. <https://doi.org/10.3390/buildings12050685>
- Santhanavanich, T., Coors, V., 2021. CityThings: An integration of the dynamic sensor data to the 3D city model. *Environment and Planning B: Urban Analytics and City Science* 48, 417–432. <https://doi.org/10.1177/2399808320983000>
- Santhanavanich, T., Padsala, R., Rossknecht, M., Dabirian, S., Saad, M., Eicker, U., Coors, V., 2023. ENABLING INTEROPERABILITY OF URBAN BUILDING ENERGY DATA BASED ON OGC API STANDARDS AND CITYGML 3D CITY MODELS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* X-1/W1-2023, 97–105. <https://doi.org/10.5194/isprs-annals-X-1-W1-2023-97-2023>
- Santhanavanich, T., Padsala, R., Würstle, P., Coors, V., 2022. THE SPATIAL DATA INFRASTRUCTURE OF AN URBAN DIGITAL TWIN IN THE BUILDING ENERGY DOMAIN USING OGC STANDARDS. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* X-4/W2-2022, 249–256. <https://doi.org/10.5194/isprs-annals-X-4-W2-2022-249-2022>
- Santhanavanich, Thunyathep, Sihombing, R., Kabiro, P.M., Würstle, P., Sini, S.K., 2022. Storing and Visualising Dynamic Data in the Context of Energy Analysis in the Smart Cities, in: Coors, V., Pietruschka, D., Zeitler, B. (Eds.), *iCity. Transformative Research for the Livable, Intelligent, and Sustainable City: Research Findings of University of Applied Sciences Stuttgart*. Springer International Publishing, Cham, pp. 251–265. https://doi.org/10.1007/978-3-030-92096-8_16
- Scalas, A., Cabiddu, D., Mortara, M., Spagnuolo, M., 2022. Potential of the Geometric Layer in Urban Digital Twins. *ISPRS International Journal of Geo-Information* 11, 343. <https://doi.org/10.3390/ijgi11060343>
- Schaufler, C., Schwimmer, E., 2020. City Information Modeling – an expedient tool for developing sustainable, responsive and resilient cities? *IOP Conf. Ser.: Earth Environ. Sci.* 588, 032005. <https://doi.org/10.1088/1755-1315/588/3/032005>
- Schrotter, G., Hürzeler, C., 2020. The Digital Twin of the City of Zurich for Urban Planning. *PFG – Journal of Photogrammetry Remote Sensing and Geoinformation Science* 88. <https://doi.org/10.1007/s41064-020-00092-2>
- Sepasgozar, S.M.E., 2021. Differentiating Digital Twin from Digital Shadow: Elucidating a Paradigm Shift to Expedite a Smart, Sustainable Built Environment. *Buildings* 11, 151. <https://doi.org/10.3390/buildings11040151>
- Shahat, E., Hyun, C.T., Yeom, C., 2021. City Digital Twin Potentials: A Review and Research Agenda. *Sustainability* 13, 3386. <https://doi.org/10.3390/su13063386>
- Shahzad, M., Shafiq, M.T., Douglas, D., Kassem, M., 2022. Digital Twins in Built Environments: An Investigation of the Characteristics, Applications, and Challenges. *Buildings* 12, 120. <https://doi.org/10.3390/buildings12020120>
- Song, J., Le Gall, F., 2023. Digital Twin Standards, Open Source, and Best Practices, in: Crespi, N., Drobot, A.T., Minerva, R. (Eds.), *The Digital Twin*. Springer International Publishing, Cham, pp. 497–530. https://doi.org/10.1007/978-3-031-21343-4_18
- Souza, L., Bueno, C., 2022. City Information Modelling as a support decision tool for planning and management of cities: A systematic literature review and bibliometric analysis. *Building and Environment* 207, 108403. <https://doi.org/10.1016/j.buildenv.2021.108403>

- Stoter, J.E., Arroyo Otori, G.A.K., Noardo, F., 2021. Digital Twins: A Comprehensive Solution or Hopeful Vision? GIM International: the worldwide magazine for geomatics 2021.
- Stoter, J.E., Otori, G.A., Dukai, B., Labetski, A., Kavisha, K., Vitalis, S., Ledoux, H., 2020. State of the art in 3D city modelling: Six challenges facing 3D data as a platform. GIM International: the worldwide magazine for geomatics 34.
- Stouffs, R., Tauscher, H., Biljecki, F., 2018. Achieving Complete and Near-Lossless Conversion from IFC to CityGML. IJGI 7, 355. <https://doi.org/10.3390/ijgi7090355>
- Tan, Y.R., Hofmeister, M., Phua, S.Z., Brownbridge, G., Rustagi, K., Akroyd, J., Mosbach, S., Bhave, A., Kraft, M., 2025. Beyond connected digital twins – Can digital twins really deliver sustainable cities? Sustainable Cities and Society 131, 106596. <https://doi.org/10.1016/j.scs.2025.106596>
- Tao, F., Zhang, H., Liu, A., Nee, A., 2019. Digital Twin in Industry: State-of-the-Art. IEEE Transactions on Industrial Informatics 15, 2405–2415. <https://doi.org/10.1109/TII.2018.2873186>
- Tomko, M., Winter, S., 2019. Beyond digital twins – A commentary. Environment and Planning B: Urban Analytics and City Science 46, 395–399. <https://doi.org/10.1177/2399808318816992>
- Tzachor, A., Sabri, S., Richards, C.E., Rajabifard, A., Acuto, M., 2022. Potential and limitations of digital twins to achieve the Sustainable Development Goals. Nat Sustain 5, 822–829. <https://doi.org/10.1038/s41893-022-00923-7>
- van der Vaart, J., Stoter, J., Diakit , A., Biljecki, F., Otori, K.A., Hakim, A., 2024. Assessment of the LoD Specification for the Integration of BIM-Derived Building Models in 3D City Models, in: Kolbe, T.H., Donaubauer, A., Beil, C. (Eds.), Recent Advances in 3D Geoinformation Science. Springer Nature Switzerland, Cham, pp. 171–191. https://doi.org/10.1007/978-3-031-43699-4_11
- van der Vaart, J.A.J., Arroyo Otori, G.A.K., Stoter, J.E., 2025. BIM2Geo converter. Delft University of Technology, Faculteit Bouwkunde, Delft.
- VanDerHorn, E., Mahadevan, S., 2021. Digital Twin: Generalization, characterization and implementation. Decision Support Systems 145, 113524. <https://doi.org/10.1016/j.dss.2021.113524>
- Virtanen, J.-P., Jaalama, K., Puustinen, T., Julin, A., Hyyp , J., Hyyp , H., 2021. Near Real-Time Semantic View Analysis of 3D City Models in Web Browser. ISPRS International Journal of Geo-Information 10, 138. <https://doi.org/10.3390/ijgi10030138>
- Weil, C., Bibri, S.E., Longchamp, R., Golay, F., Alahi, A., 2023. A Systemic Review of Urban Digital Twin Challenges, and Perspectives for Sustainable Smart Cities. Sustainable Cities and Society 104862. <https://doi.org/10.1016/j.scs.2023.104862>
- W rstle, P., Padsala, R., Santhanavanich, T., Coors, V., 2022. VIABILITY TESTING OF GAME ENGINE USAGE FOR VISUALIZATION OF 3D GEOSPATIAL DATA WITH OGC STANDARDS. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. X-4/W2-2022, 281–288. <https://doi.org/10.5194/isprs-annals-X-4-W2-2022-281-2022>
- W rstle, P., Santhanavanich, T., Padsala, R., Coors, V., 2020. The Conception of an Urban Energy Dashboard using 3D City Models. pp. 523–527. <https://doi.org/10.1145/3396851.3402650>
- Wysocki, O., Schwab, B., Beil, C., Holst, C., Kolbe, T.H., 2024. Reviewing Open Data Semantic 3D City Models to Develop Novel 3D Reconstruction Methods. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLVIII-4-2024, 493–500. <https://doi.org/10.5194/isprs-archives-XLVIII-4-2024-493-2024>
- Xia, H., Liu, Z., Efremochkina, M., Liu, X., Lin, C., 2022. Study on city digital twin technologies for sustainable smart city design: A review and bibliometric analysis of

- geographic information system and building information modeling integration. *Sustainable Cities and Society* 84, 104009. <https://doi.org/10.1016/j.scs.2022.104009>
- Xiong, M., Wang, H., 2022. Digital twin applications in aviation industry: A review. *Int J Adv Manuf Technol* 121, 5677–5692. <https://doi.org/10.1007/s00170-022-09717-9>
- Xu, H., Omitaomu, F., Sabri, S., Zlatanova, S., Li, X., Song, Y., 2024. Leveraging Generative AI for Urban Digital Twins: A Scoping Review on the Autonomous Generation of Urban Data, Scenarios, Designs, and 3D City Models for Smart City Advancement.
- Xu, Z., Qi, M., Wu, Y., Hao, X., Yang, Y., 2021. City Information Modeling: State of the Art. *Applied Sciences* 11, 9333. <https://doi.org/10.3390/app11199333>
- Xue, F., Lu, W., Chen, Z., Webster, C.J., 2020. From LiDAR point cloud towards digital twin city: Clustering city objects based on Gestalt principles. *ISPRS Journal of Photogrammetry and Remote Sensing* 167, 418–431. <https://doi.org/10.1016/j.isprsjprs.2020.07.020>
- Yan, J., Lu, Q., Li, N., Chen, L., Pitt, M., 2025. Common data environment for digital twins from building to city levels. *Automation in Construction* 174, 106131. <https://doi.org/10.1016/j.autcon.2025.106131>
- Yossef Ravid, B., Aharon - Gutman, M., 2022. The Social Digital Twin: The Social Turn in the Field of Smart Cities. *Environment and Planning B: Urban Analytics and City Science* 239980832211370. <https://doi.org/10.1177/23998083221137079>
- Zhang, X., Shen, J., Saini, P.K., Lovati, M., Han, M., Huang, P., Huang, Z., 2021. Digital Twin for Accelerating Sustainability in Positive Energy District: A Review of Simulation Tools and Applications. *Frontiers in Sustainable Cities* 3.
- Zheng, X., Lu, J., Kiritsis, D., 2022. The emergence of cognitive digital twin: vision, challenges and opportunities. *International Journal of Production Research* 60, 7610–7632. <https://doi.org/10.1080/00207543.2021.2014591>
- Zhu, J., Wu, P., 2022. BIM/GIS data integration from the perspective of information flow. *Automation in Construction* 136, 104166. <https://doi.org/10.1016/j.autcon.2022.104166>
- Zhu, Q., Hu, M., Zhang, Y., Du, Z., 2009. Research and practice in three-dimensional city modeling. *Geo-spat. Inf. Sci.* 12, 18–24. <https://doi.org/10.1007/s11806-009-0195-z>

CURRICULUM

Imane Jeddoub was born in Rabat, Morocco, in 1998. She obtained a State Engineer degree in Land Surveying and Geomatics from the Institut Agronomique et Vétérinaire Hassan II (IAV Hassan II), Rabat, in 2021. Her strong interest in geomatics research motivated her to pursue advanced studies in this field. In late 2021, she completed a research internship in Germany at the Helmholtz Institute Freiberg for Resource Technology (HIF-HZDR), where she worked on her final engineering project.

In March 2022, she joined the GeoScITY Laboratory (formerly the Geomatics Unit) at the University of Liège, Belgium, as a PhD student. Her PhD position was funded by the Fonds de la Recherche Scientifique (F.R.S.-FNRS) within the framework of a PDR research project, under the supervision of Prof. Roland Billen. Her research focuses on the design and implementation of integration levels in Urban Digital Twins (UDTs), in alignment with the main objectives of the PDR project.

This research is part of the GIS 3.0 project, which investigates the convergence between Geographic Information Systems and Web 3.0 technologies, including Semantic Web approaches, object-oriented prototype languages (such as JavaScript and JSON), and document-oriented NoSQL databases. The project is funded by the Belgian National Fund for Scientific Research (FNRS_2019_SIG3.0_PDR/OL T.0024.20).

Alongside her PhD research, she has taken on scientific roles in several applied projects and research proposals related to Urban Digital Twins and geoinformation. She is the scientific lead of the City2Twin platform and actively contributes to applied projects targeting industrial use cases, such as WaLoD2, which focuses on the 3D reconstruction of buildings at regional scale in Wallonia, Belgium. She collaborates with colleagues across the geospatial data pipeline, including data processing, management, and dissemination.

She has also contributed to teaching activities at the University of Liège, supporting courses in R&D in Geoinformation, FME, and Urban GIS, and co-supervising MSc theses. She has served as a reviewer for international journals, including Building and Environment, Computers, Environment and Urban Systems, and Sustainable Cities and Society. She maintains strong links with international research communities through collaborations and participation in scientific conferences.

Since February 2026, she has been a researcher at the GeoScITY Laboratory, leading the development of the City2Twin platform and related Urban Digital Twin projects.

LIST OF PUBLICATIONS

International Journal Papers (ISI)

Jeddoub, I., Nys, G.-A., Hajji, R., & Billen, R. Digital Twins for cities: Analyzing the gap between concepts and current implementations with a specific focus on data integration. *International Journal of Applied Earth Observation and Geoinformation*, 122, 103440. doi:10.1016/j.jag.2023.103440

Jeddoub, I., Nys, G.-A., Hajji, R., & Billen, R. Data integration across urban digital twin lifecycle: a comprehensive review of current initiatives. *Annals of GIS*, 1-20. doi:10.1080/19475683.2024.2416135

International Refereed Conferences

Jeddoub, I.*, Ballouch, Z.*, Hajji, R., & Billen, R. (2024). Enriched Semantic 3D Point Clouds: An Alternative to 3D City Models for Digital Twin for Cities? In T. H. Kolbe, A. Donaubauer, ... C. Beil (Eds.), *Recent Advances in 3D Geoinformation Science - Proceedings of the 18th 3D GeoInfo Conference*. Springer Science and Business Media Deutschland GmbH. doi:10.1007/978-3-031-43699-4_26

* These authors have contributed equally to this work.

Ballouch, Z., **Jeddoub, I.**, Hajji, R., Kasprzyk, J.-P., & Billen, R. Towards a Digital Twin of Liege: The Core 3D Model based on Semantic Segmentation and Automated Modeling of LiDAR Point Clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4/W4-2024, 13 20. doi:10.5194/isprs-annals-x-4-w4-2024-13-2024

Rafamatanantsoa, B. P., **Jeddoub, I.**, Yarroudh, A., Hajji, R., & Billen, R. City2Twin: an open urban digital twin from data integration to visualization and analysis. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-2/W8-2024, 387-394. doi:10.5194/isprs-archives-xlvihi-2-w8-2024-387-2024

Doi, H., Yarroudh, A., **Jeddoub, I.**, Hajji, R., & Billen, R. Automatic Detection and 3D Modeling of City Furniture Objects using LiDAR and Imagery Mobile Mapping Data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-2/W8-2024, 125-132. doi:10.5194/isprs-archives-xlvihi-2-w8-2024-125-2024

Kharroubi, A., Ballouch, Z., **Jeddoub, I.**, Hajji, R., & Billen, R. Automated detection and structuration of building and vegetation changes from LiDAR point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-2/W8-2024, 227-233. doi:10.5194/isprs-archives XLVIII-2-W8-2024-227-2024

Yarroudh, A., Kharroubi, A., **Jeddoub, I.**, Ballouch, Z., & Billen, R. Railway reconstruction from 3D point cloud using Deep Learning and Parametric Modeling. *International*

Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVIII-2/W8-2024, 477-482. doi:10.5194/isprs-archives-xlvi-2-w8-2024-477-2024

Yarroudh, A., Kharroubi, A., **Jeddoub, I.**, & Billen, R. Automatic upgrade of 3D building models to LoD3 using 3D Point Clouds and Grounding DINO. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVIII-2/W8-2024, 471-476. doi:10.5194/isprs-archives-xlvi-2-w8-2024-471-2024

Jeddoub, I., Roquet, M., Oudadda, Y., Hajji, R., Dewallef, P., & Billen, R. A Lightweight Framework for Seamless Integration of Building Energy Simulations into Urban Digital Twins. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVIII-4/W15-2025, 77-84. doi:10.5194/isprs-archives-xlvi-4-w15-2025-77-2025

Macq, H., **Jeddoub, I.**, & Billen, R. Beyond Twinness and Twinning: Reflections on Urban Digital Twinnation from the Namur 3D Project. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVIII-4/W16-2025, 51-58. doi:10.5194/isprs-archives-xlvi-4-w16-2025-51, 2025

Doi, H., Hajji, R., **Jeddoub, I.**, and Billen, R.: Automatic Detection, 3D Localization, and Semantic Enrichment of Commercial Signboards Using 360° Mobile Mapping Imagery: A Case Study in Temara, Morocco, ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., X-4/W6-2025, 73-80, <https://doi.org/10.5194/isprs-annals-X-4-W6-2025-73-2025>, 2025.