TOWARDS AN ONTOLOGY FOR THE STRUCTURING OF REMOTE SENSING OPERATIONS SHARED BY DIFFERENT PROCESSING CHAINS

G.-A. Nys[1] , J.-P. Kasprzyk[1], P. Hallot[2], R. Billen[1]

[1] Geomatics Unit, Department of Geography, ULiège - Place du 20 Août, 4000 Liège (ganys/JP.Kasprzyk/rbillen)@uliege.be
[2] LNA-DIVA, Faculty of Architecture, ULiège - Boulevard de la Constitution, 41 4020 Liège p.hallot@uliege.be

Commission IV, WG IV/2

KEY WORDS: Ontology engineering, Remote sensing, NoSQL database, Graph mining, Knowledge discovery

ABSTRACT:

This paper proposes an ontology to structure and describe processing chains in the remote sensing field. These chains are made up of elementary elements (operations) organized in collections. The collection notion, including information about order and repeatability of the elements, is widely defined by using the relations between their constituting items and relations to the whole data store. Applications of the ontology are illustrated with web services provided by a platform for users and providers of processing chains. A graphical interface facilitates data integration in a RDF triple store. Thanks to the management of metadata (ISO19115-3), relevant information can be requested by intelligent search engines. Graph analysis, errors management and consistency rules are computed in order to gather coherent information from the different sources. Results of these analyses are then used by machine learning algorithms for new knowledge discovery.

1. INTRODUCTION

Nowadays, remote sensing is increasingly used in administrations and private sector. The pooling of knowledge engaged with the apparition of web platforms dedicated in remote sensing increases. Such platforms are used both by users for specific services retrieval and by providers to publish tools and applications.

All the difficulty in such a use lies in the fact that not only the conceptualization but also the terms definitions needs to be considered with the most relevant vocabulary according to the users and the providers. This vocabulary needs to be consistent and leaves as little openness as possible to interpretations. A unique vocabulary needs to be set to facilitate comparison, retrieval and publishing in these platforms.

The remainder of this paper is structured as follows. First, we develop the methodology to create and manage the creation of a formal ontology dedicated to the structuring of processing chains in remote sensing. Then, the different considerations that lead to the graph structure are depicted. Consistency analysis and rules are the next logical step. The development of a user-friendly interface favours the use of the ontologies, which are usually restricted to specialized staff. Creation, modification and removal of instances in the database are also available and this is a step further for ontologies integration in many tools. Finally, these considerations are illustrated through the creation of a user-friendly interface and knowledge graph mining.

2. STATE OF THE ART

The definition in a formalized and well-documented structure of the different objects allows the comparison of the different elements, as for instance in social graphs (Braun, Cuzzocrea, Leung, Pazdor, and Tran, 2016). In computer science terms (as it pertains to knowledge graphs) an ontology formally describes the types, properties and interrelationships between entities of real-world concepts. In other words it is „an explicit specification of a conceptualization" (Gruber, 1993). Those are

usually used to restrict elaboration and organize data into information than into knowledge, as defined in (Rowley, 2007).

Once the ontology implemented in a NoSQL graph database, data can also be processed for business intelligence and new knowledge discovery in databases (KDD) purposes (Fayyad, Piatetsky-Saphiro, and Smyth, 1996).

Knowledge discovery faces long runtime and difficulties to compute efficient mining algorithm (Han and Kamber, 2012; Miller and Han, 2009). Ontologies are usually used as support material to guide such a retrieval (Gómez-Pérez, Fernández-López, and Corcho, 2010; Inokuchi, Washio, and Motoda, 2000), especially for geographic databases (Bogorny, Engel, and Alvares, 2007). In biomedical studies, where ontologies have firstly been used, semantic patterns retrieval provide meaningful structure analysis (Hilario, Nguyen, Do, Woznica, and Kalousis, 2011; Huang, Dou, He, Hayes, and Dang, 2010; Shen and Lee, 2016).

The use of ontologies also allows the merging of the knowledge base described by the ontology with other knowledge bases (Noy and Musen, 2003; Stumme and Maedche, 2001). It is part of the pooling of knowledge started in Linked Open Data. Moreover, data are computable by other projects and reusable in different contexts.

In remote sensing, ontologies are usually used to support entities classification or image segmentation (Andrés, Arvor, Mougenot, Libourel, and Durieux, 2017; Andres, Arvor, and Pierkot, 2012). Many works have presented different conceptualizations and hierarchies to describe the links between the entities in images (Oliva-Santos, Maciá-Pérez, and Garea-Llano, 2014). Satellite imagery metadata are also structured in ontologies to support search engine (Lin, Xu, and Bai, 2017).

Creating a new ontology is a time-consuming and complex process. A rigorous methodology is thus necessary. Most of the time, such ontologies are created by knowledge engineers, who are familiar with the tools and standard hypothesis in ontologies conception. However, these are usually not experts in the

application domain covered by the ontology. It is therefore needful to reduce this gap by designing the most relevant conceptualization (Denaux, Dolbear, Hart, Dimitrova, and Cohn, 2011).

Moreover, despite the ontology creation for a dedicated application, it is important not to forget that the invested efforts are commonly destined to be reused, enhanced and shared. It is important to consider the correctness, applicability and usability during the creation of such an ontology. Protégé is a tool particularly suited for the evaluation and the creation of an ontology in the most complete way (Tan, Adlemo, Tarasov, and Johansson, 2017).

The description of processing chains is a common preoccupation and many solutions have already been proposed in the literature. Several projects propose a semi-automatic creation of processing chains for web services using semantic web technologies (Sirin, Hendler, and Parsia, 2003; Srivastava and Koehler, 2003). The building of such services is based on the concatenation of pre-existing subservices linked together by inputs and outputs data. It is a simple but relevant chain.

The next logical phase needs this step to be automatized and consistency checked. Indeed, rules for sharing and consistency analysis are crucial to favour interoperability of concepts and their understanding (Grau, Horrocks, Kazakov, and Sattler, 2008). The consistency will be translated in adapted rules set (Yue, Di, Yang, Yu, and Zhao, 2007).

Finally, artificial intelligence, neural networks as an example, may make a step further in the automatized creation of processing chains and their complexity. Liu, Xue, Guang, and Liu (2015) state the possibility to make such a proposition in remote sensing.

3. RESEARCH HYPOTHESIS

According to the state of the art, the creation of a formal ontology structuring the proposition of services may improve the creation of web platforms dedicated to remote sensing. Expressing the different services propositions (i.e. processing chains) in a common and well described formalism permits the an easy exchange between users and providers.

Implementing such an ontology in a NoSQL graph database and providing applications on this basis is also an enhancement that leads to comparison, error detection and automating of knowledge discovery.

4. APPLICATION ONTOLOGY

In its conceptual version, an ontology is represented as an oriented graph. The ontology created in this study is a hybrid one, made up of two different sub-ontologies, as illustrated in Fig. 1. One is used to define the various types of collections, and consequently structure the processing chains. The other describes the vocabulary and concepts in remote sensing: operations, data and services.

The first part (see section 4.1) will be illustrated through the instantiation and allows the definition of processing chains, the first part (see section 4.2) is a higher-level ontology used to set rules about the different subclasses.

What are the benefits of this methodology and its implementation in a NoSQL database? The description of each

service is independent of its level of detail and completeness. The absence of information does not need to result of a "No Data" value but only the absence of the attribute in the graph. This results in a more suitable management of memory usage and scalability in users' experience.

The instanced ontology and instance-clean ontology are hosted at: http://www.purl.org/net/eor_ontology. All the instances have been declared following the *eor:* prefix namespace: http://www.geo.ulg.ac.be/nys/. It will be used in this article as notation simplification.
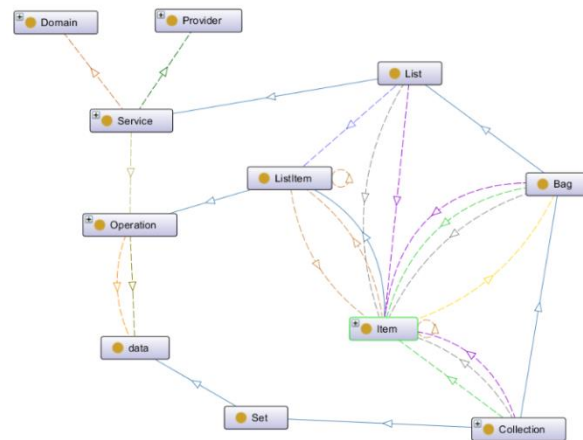

Figure 1. Visualisation of the ontology structure via OntoGraf Plugin in Protégé

4.1 Classes definition

There are three main classes besides domain and provider, which both do not need any explanation: Data, Operations and Service. These are the constituting elements of the processing chains.

4.1.1 eor:Data: It is customary for data to be described by metadata. The creation / modification of that metadata is an elementary operation. The owner of a data is, besides other characteristics, not modified in a processing chain. This kind of metadata is designated as "not operable" and is therefore not part of this study, unlike the spatial resolution for example.

On the contrary, several metadata are defined as "operable" and are the result of an elementary operation, like spatial resolution. An elementary operation is thus an operation that creates or modifies at least one information.

On the one hand, a data is used as input by an operation. Based on this kind of relation, restrictions arise on the data format. This is detailed in the consistency rules section. On the other hand, the result(s) of an operation is (are) also a data. The modification is made on the metadata but not on the nature of the element, which is still a data.

Finally, a collection of data, structured or not, is still is a data. The difference between both the collection and the data themselves is the creation of metadata related to the idea of collection (index, temporal resolution between successive elements, *etc*.). See section 4.5 for description of the collection concept.

4.1.2   eor:Operation: As mentioned above, an elementary operation creates at least one information based on the input data and results in the creation of a new data, the output. The knowledge about an operation allows the analysis of its influence on the metadata. Thus, it is possible to determine the consistency analysis and quality management of a processing chain by splitting it in successive elementary operations.

An operation can be of two natures:

- Algebraic operation. This kind of operation consists in the modification of at least one metadata. In this operations family stand filters, normalization, resampling, interpolations, map algebra, spatial aggregations … In other words, the answer to an algebraic operation can be provided by an indisputable formula or algorithm.

- Descriptive operation. This kind of operations leads to the creation of information but without the modification of the initial data. It is typically the case when it is talked about classification, count … This type of operation is often subject of human expertise and it therefore hardly appreciable and the result may differ from the operators.

Although these conceptual differences may provide a good hierarchy, the distinction between these two operation families is not relevant in the model. Thus, the operation class is generic and despite the fact that we consider remote sensing, the notion of elementary operations is independent of the application field.

Rather to structure the different subclasses of operations, the hierarchy has been selected from the Orfeo Toolbox[1]. Each operation instance is therefore defined as an instance of its subclass type. Every subdivision is disjoint from the others.

The notion of operations collection is not as simple as the data one. An operation is defined as the creation/modification of an information on a data. However, because of the nature of these changes, precedence and necessary relations appear. Many operations inputs need to be formalised in a particular file extension, scaled on a specific reference standard or georeferenced for instance.

At a second level, a more detailed classification of operations is requisite. Different specializations are thus defined: their definitions are more precise and allow a more relevant management. It is also at this level of definition that the real extensions of the conceptual classes are specified. Those are the individuals.

4.1.3   eor:Service: The service class is defined as a collection of operations where the order has an importance. Operations constituting a service are an ordered list where elements are repeatable. The notion of order is independent of any notion of index. Indeed, an operation may appear multiple times in a service and the relations between the service and its operations are not functional (cardinality one-to-one).

Each service is defined by a description following the Dublin Core prescriptions (dc:description). A service is also described by linking two Classes with ObjectProperties:

- eor:Domain: This class represents the main thematic field of a processing chain. As an example, agriculture and subsidence are

two applications domain in remote sensing. The corresponding ObjectProperty is eor:hasDomain.

- eor:Provider: this class binds a processing chain to a provider. Based on this link, a provider can be defined not only by his name but by his economic model, certification process … The corresponding ObjectProperty is eor:hasProvider.

4.2   Collection concept

As explained before, the notion of collection is an important issu. Collections structure services and describe the arrangement between their constituting elements: the operations. Nevertheless, collections of data are also customary and consequently need to be managed. Two solutions have been investigated:

- A simple model defining the collection concept as a reflexive relation from the object to the object itself. The objects collection is then an object (instance) of the same class. Therefore, the modification of an instance as a collection does not bring any change to the instance, except the metadata updates (temporal resolution of the collection, number of elements …). These metadata are thus added as datatype properties on the data, which can bring fuzziness in the database.

- A more complex definition that develops the semantic definition of a collection. This can easily be achieved in a formalized ontology. The ontology defines a collection as an ordered, or not, succession of identified elements.

The definition of the simple reflexive model is naive. Certainly, it greatly facilitates the model design: a collection defines a whole whose nature is similar, even identical, to the elements that constitute it. The transitivity of such a relation is quite relevant.

However, some information, which may be crucial, is neglected: order, repeatability or the enumeration of the chains of a processing chain for example. Winston and Chaffin (Winston, Chaffin, and Herrmann, 1987) propose a taxonomy describing the different types of relations between a part and the whole-part. The considered categorization by the authors is depicted by the combinations of three characteristics:

- Functional/Non-functional: Parts are/are not in a specific spatial/temporal position with respect to each other. This supports their functional role with respect to the whole.
- Homeomerous/Non-homeomerous: Parts are similar/dissimilar to each other and to the whole (to which they belong).
- Separable/Inseparable: Parts can/cannot be physically disconnected, in principle, from the whole to which they are connected.

Based on these elements and their characteristics, different types of collections are used in our ontology:

4.2.1   Portion/Mass relation: The definition of the data collection is complex. First, the functional aspect of such collection, as defined in the previous section, is not questionable in the remote sensing field. Indeed, the relative temporal referencing of time series elements is important but this aspect is already described in the metadata. For instance, the dates of acquisition of the different images in the collection are once reported in metadata.

[1] https://www.orfeo-toolbox.org/

These elements are therefore not functional. In addition, the definition expresses the implementation of position in space and time, what is not relevant in this study. The data, as a real material object, is not applicable.

Homogeneity leaves little doubt. It is obvious that in our conception of a collection, elements are of the same nature. The quality of this homogeneity could be less or more precise. The more the level of detail precise is, the less homogeneous the collection will be.

For instance, let us take a set of images. If their definition is limited to the data format, the data sets will be recognized as similar and belonging to the raster category (in opposition to the vector category). Let us make a step further in the level of detail with the definition of the pixel type: are they coded in 1, 8 32 … bits? The first differences appear. Moreover, the number of differences between the entities increases with the precision of the different defined elements (resolution, thematic, dimensions …).

Finally, data are separable. For example, in an aggregation, typically a reduction of the temporal resolution of a time series, the number of data in the collection will decrease. Nevertheless, data are still data and their nature will not change during the operation. They are therefore separable and their existence is not linked to the notion of collection.

4.2.2 Feature/Activity relation: This type of relation directly sets the necessary notion of succession between the features in a relevant whole. It is particularly true in a service that is a collection of operations. It is necessary for a feature, operation n+1, to be preceded by another feature of the collection, operation n, whose nature is defined by the consistency of the processing chain.

Order has all its importance in the structure of the activity. The elements repeatability in this kind of relationship is not blocked if it respects the restriction previously stated. Both notions (succession and order) belong to another definition level. These cannot be considered as a simple connection. They implement a reflection and need accordingly a reasoning and additional attributes (relations of precedence and antecedence ...).

Several efforts have been made to structure such a notion of lists (Pauwels, Terkaj, Krijnen, and Beetz, 2015) and domain ontologies have been created. However, the notion of ordered list is not sufficient and too little developed in these approaches for our considerations. To this end, the approach proposed in the Collections Ontology (Ciccarese and Peroni, 2014) has been selected and imported.

This high-level ontology brings a structure for the definition of collection types and the linking of the constituents' elements between them but also to the whole. A collection is characterized by two aspects previously enounced: order and repeatability. The article defines subsequently three kinds of collections presented in Table 1. :

	Ordered	**Non-ordered**
Repeatable	Lists	Multisets/Bags
Non-repeatable	/	Sets/Sequences

Table 1. Classification of collections concepts

To this are added rules for the consistency between the different categories:

co:Set ⊑ Set
co:Bag ⊑ Bag
co:Set ⊓ co:Bag = ∅
co:List = co:Bag ⊓ Sequence

Every element, as it happens an operation, has a logical position in the collection and its repeatability can be relevant. The choice in the definition of the different collections in the Collections Ontology is thus focused on the list.

Data are non-repeatable and non-ordered. Their order is not directly specified in the ontology but tacitly in the metadata (time of acquisition …). It is not necessary to specify this arrangement in the ontology.

Moreover, the ontology introduces the concept of co:hasFirstItem and co:hasLastItem in the lists as well as the precedence and antecedence relations. Semantic Web Rule Language (SWRL) restrictions on the logic of necessity and sufficiency (specific to these relations of succession) are implemented. These relationships lead to the creation of subsumptions. SWRL is firstly used to describe rules in Description Logic then translated in OWL in a second time.

A modification is yet indispensable for the notion of collection in this approach. The collection and its constituting elements are defined as disjoint. In other terms, an entity cannot be a collection nor a collection of collections itself. It has thus been set possible that a concept defined as a collection of this concept by a reflexive relation to the collection (1…n). Thus, by transitivity of the relation from the part to the whole, a collection of collections is viable.

Caution, however, that the transitivity is permit only when the relations form the part to the whole are of the same nature. Illustrated by the following example (Winston et al., 1987) based on the previously defined relations, this affirmation takes all its sense and draws attention to the errors that can be encountered by a gross syllogistic reflection.

1.   *Simpson's arm is a part of Simpson.*
2.   *Simpson is a part of the Philosophy Department.*
3.   *Simpson's arm is a part of Philosophy Department.*

It is dangerous to consider that the third affirmation is true based on the two previous. The example here is trivial for a human: Simpson's arm is obviously not a full member of the Department of Philosophy. Nevertheless, the last affirmation may seem direct and easy for a machine. Humans perceive the differences but computers need to be trained and relations need to be well built. It is important to respect the homogeneity of the meronymic relations to keep coherence and understanding.

About this consideration, transitivity is not possible between the different types of collections in this study since corresponding to different relations of the part to the whole. Just as the arm of Simpson cannot be part of the Philosophy Department. Thus, the transfer of knowledge on different levels of elements is prevented and the entities disjoint.

4.3 Inference and error management

Reasoning on relations and classes hierarchy can lead to discover information in knowledge databases. The inferred

knowledge bases thus allow creating new statements and detecting errors in the asserted graph. OWL has description logic based language allows reasoning to infer logical statements from a set of asserted facts or axioms.

Asserted facts are depicted in both the structure ontology (terminological box - TBox) and its individuals (assertional - ABox) boxes. In other words, TBox is constituted of classes definitions in the EOR ontology. Thanks to edited rules and restrictions on classes definitions, those boxes are inferred and thus structure the knowledge discovered on logical reasoning. Listing X provide a simple example of inference on both boxes. An ontology presenting errors or incoherence is defined as inconsistent.

The inconsistency is translated by the blocking of all reasoning. It would be interesting to be able to overcome this limitation and thus permit correction on the wrong-inferred triples. It would then be accepted that the use of the ontology is not optimal because at least an error remains. The general process is then available and can follow is course without being frozen.

This trick is made possible because of the integration of a Datatype Property:hasError . This association is a relation linking an Individual to a characters chain, as it happens in this study: the error message explaining the presence of error.

The appearance of an error message is defined by consistency rules expressed in SWRL. Thus, when an error happens, it will suffice to query the presence of such a property without blocking the use of the knowledge base.

4.4 Consistency rules

A database can contain contradictions. These contradictions can be, in the case of ontologies, the results of bad reasoning on inferred information. It is important to keep a logical consistency when it comes to description logic and tableau based algorithm. Pellet (Sirin, Parsia, Grau, Kalyanpur, and Katz, 2007) has been selected as reasoning engine especially because of its support of SWRL built-ins.

 Logical consistency is one of the six elements defining the data quality as depicted in ISO 19157:2013. This definition divides the consistency into four sub-concepts as following: conceptual consistency, coherence of value domain, format consistency and topological coherence.

Conceptual consistency is characterized by the intrinsic ontology definition and their constituting elements themselves. It is the responsibility of the knowledge engineer and therefore, especially needs the support of domain experts.

The coherence of value domain are described in domains/ranges of the relations. Ontologies also give the possibility to set up cardinality, reflexivity, transitivity … for the relations. These elements define the coherence of value domain also. Format consistency is described in general axioms limiting the use of data in the different services. In this study, this kind of coherence is set up in the general axioms.

Topological coherence is not relevant in this study because of the ontology nature itself. Relations therefore guaranty the integrity of topology, which is already discussed in the conceptual consistency.

4.4.1 General axioms: General axioms translate base principles of remote sensing but also rules applied to collections types. Considering the Open World Assumption, it is easier to define inconsistency rules rather than consistency rules. An incoherence will lead to the creation of an error. What are the conditions that lead to an error? The general axiom defining the incoherence of errors presence states that the number of errors need to be zero.

During the reasoning, whenever an error is created (the entity then has an Object Property hasError), the description logic engine flags an error and gives its explanation. It is then considered as inconsistent that an entity has an error as property.

As already said, the Collections Ontology is used to define the notion of collection but also to limit the range of the classes. Many rules in this ontology are general axioms governing the whole graph. Despite the trivial nature of these rules, it is necessary to precise them because of the Open World Assumption.

1. co:Bag and (co:hasItem only co:ListItem) SubClassOf co:hasItem only co:ListItem
2. co:List and (co:hasFirstItem some co:Item) and (co:hasLastItem some co:Item) SubClassOf co:hasLastItem some co:Item
3. co:List and (co:hasItem some co:ListItem) SubClassOf co:List and (co:hasFirstItem some co:Item) and (co:hasLastItem some co:Item)
4. co:ListItem and (co:hasNextItem exactly 0 owl:Thing) SubClassOf co:hasNextItem exactly 0 owl:Thing
5. co:List and (co:hasItem some co:ListItem) SubClassOf co:hasLastItem some co:Item
6. co:List and (co:hasItem some co:ListItem) SubClassOf co:hasItem some co:ListItem

Table 2 SWRL Rules on Collection Ontology

4.5 Operations definition

Operations are defined as the constituting elements of the processing chains, as co:Item of a co:List. The inconsistency rules are therefore built around these.

Rule: co:hasPreviousItem(?x, ?y), eor:hasInput(?x, ?a), eor:hasOutput(?y, ?b), owl:differentFrom(?a, ?b) -> eor:hasError(?x, "Error : The I/O Data are not valid."^^xsd:string)

In this context, consistency is established by the integrity of the data exchanged between two successive operations in a processing chain. It is necessary that an output of the n operation is equal to at least one input of the n+1 operation.

Thus, if the I/O of two successive operations do not match, an error is created for the second operation. This allows cutting the processing chain after the first operation and managing the error at the source.

Rule: co:hasPreviousItem(?x, ?x) -> eor:hasError(?x, "Error : An item is followed by itself."^^xsd:string)

In this study, an element of a collection cannot be followed by itself. For the operations collection, it is obvious that an operation cannot be done twice in a row.

4.5.1 **Spatial reference identifier:** The introduction of the Spatial Reference Identifier (SRID) in the ontology brings tacitly rules. This identifier is used to describe data using a Datatype Property. This key-value defines many uses and restrictions:

- The geographic coordinates are expressed in degrees (decimals, sexagesimal or degrees minutes seconds, degrees minutes).
- The projected coordinates are expressed in meters.

The SRID is unique: the Datatype Property is therefore functional (the number of relation is limited to one). Additionally, it is specified that the datatype has to be an integer. CRS (Coordinates Reference Systems) transformations and definitions are the only operations that change the SRID. These exceptions lead to a rule that expresses an error if the SRID has been changed and the operation is not of "Projection" type (transformation or definition).

Rule: eor:hasInput(?x, ?a), eor:hasOutput(?x, ?b), eor:hasSRID(?a, ?aSRID), eor:hasSRID(?b, ?bSRID), swrlb:notEqual(?aSRID, ?bSRID), (not (Projection))(?x) -> eor:hasError(?x, "Error : The I/O Data are not valid."^^xsd:string)

5. KNOWLEDGE DISCOVERY

The representation of knowledge as a graph can lead to new knowledge discovery or even used as knowledge base for machine learning algorithms. The subgraph describing the operations hierarchy can be more detailed and permit graph mining and services comparisons (patterns recognition, "hotspots" highlighting …). Semantic patterns are so the constituting elements that we focus on.

To be able to determine some consistency from the structured graph can lead to rules supporting artificial intelligence reasoning. It would then be possible to compare these restrictions with human expertise. Some rules, trivial for a human, may be transcript within the data. Therefore, it will lead to edit an expressive rule for machines understanding. One of those tacit rules may set succession between operations. The provided example studied the actual services proposition to determine succession guidelines:

As it has been defined, services are considered as co:List, which are a specialisation of co:Bag. Services bags are neither more nor less than unordered sets of elements S. In order to compare the services between them, Jaccard's matrix J has been built. Each element $J(i, j)$ of the triangular matrix is computed as the Jaccard's index, which is the ratio of the intersection over the union of the two operations sets, service S_i and service S_j. The ratio is comprised between 0 and 1 both include and is computed as following:

$$J(i,j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \qquad (1)$$

Similarities become relevant only when operations types are concerned. It is important to compare the operations types but not the operations themselves. Indeed, instantiation can be made by different applications or operators. This thus leads to heterogeneity in the data and will not trigger any similarities detection. This upper-level of definition is therefore relevant as shown on Table 1.

	UNP03	ISP04	UNP02	SeF04
UNP03	100.00	024.44	087.14	042.86
ISP04	024.44	100.00	020.00	037.50
UNP02	087.14	020.00	100.00	012.50
SeF04	042.86	037.50	012.50	100.00

Table 3 Extract of the Jaccard index matrix

This approach considers that a service is an unordered bag of operations. However, order can have an influence on the comparison of services and lead to different rules. To study the influence of order in the correlation, another algorithm has been set. An ontology is a semantic expressive directed graph. It means that relations are directed and it is possible to extract only the relations of antecedences between the operations.

Collections Ontology specify that a co:List is composed of co:Item, which are linked by relation co:hasNextItem and its inverse relation co:hasPreviousItem. By browsing the triples and extracting the subjects and objects of these relations, it is possible to determine an occurrence index. An occurrence of 100% means that, in the provided processing chains, an operation type, the subject, is always followed by the same operation type, the object. This indicator is then used to set rules for machine learning support and detect errors by reasoning on the knowledge base. Table 4. provides an extract of result, for the specific operation type "Stacking":

1.	1 | 0.11 | Stacking	follows	DataPreProcessing
2.	1 | 0.11 | Stacking	follows	SARPreProcessing
3.	2 | 0.22 | Stacking	follows	ImageManipulation
4.	5 | 0.56 | Stacking	follows	FeaturesExtraction

Table 4 Code sample of Stacking operations type

In the scope of the project, twenty-one full-described processing chains have been developed and structured in the triple store. The algorithms at best give 71% of occurrences between two successive operations types, which is not sufficient to edit an axiom. It is planned to work on bigger datasets to validate the methodology.

6. GRAPHICAL USER INTERFACE

Ontologies are great to structure domain knowledge but they are time-consuming and need both an ontology engineer and a domain expert (Denaux et al., 2011). However, discussions between the two parties often turn into misunderstandings and introduce errors, omissions …
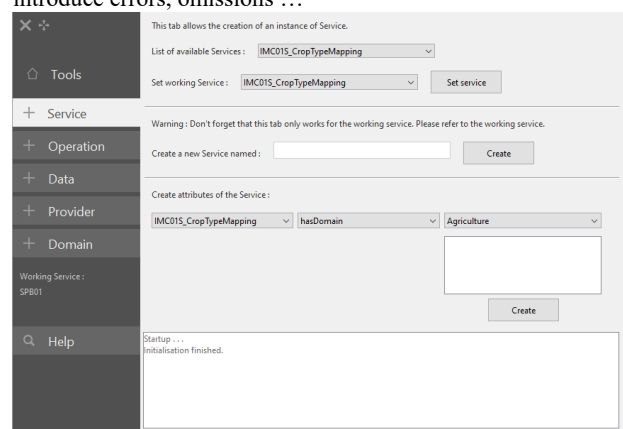

Figure 2. Graphical User Interface proposition

Even if the creation and the management of ontologies are now becoming more and more controlled, the proliferation of instances is also an important aspect to manage (Stefanidis, Chrysakis, and Flouris, 2014). When it comes to create entities in an automatic way, the quantity of instances can rapidly become a problem and an important resources issue. It is important to keep an eye on such a process and to control redundancy. Depending on the number of entities, their supervision can become very complicated.

About the data sources, information are usually imported from relational databases to the graph store using, for example, R2RML-F (Debruyne and O'Sullivan, 2016). This particular mapping allows the creation of consistent triples from the different tuples in the parent relational database.

In some applied examples, data are not created before the provision of the applications but during their use. If users have direct access to the database but are not familiar with the structure of graphs and its nomenclature, it can be useful to provide a simplified interface for database management. Such an interface can be dedicated to a particular use (automatically generated text fields, pre-set lists, etc.).

Because of non-aware users' interventions, it is also mandatory for the database security to lock the access to the servers and preserve its integrity. The tool needs thus to limit the possibilities and process verifications on the queries before interrogating the database.

Users often do not need to manage SPARQL syntax, the language dedicated to RDF Graph query, neither its subtleties. To compel these to become familiar with a new language can reduce their motivation to use new tools. The new proposed tool needs to encapsulate SPARQL queries and therefore automatize their written.

The proposed graphical user interface (See Fig. 2) is build following these considerations: simple, fluid and intuitive.
The architecture is quite traditional, besides Java Swing for the client interface:
- Ontology API 3.7.0: API to manage RDF graphs.
- TDB: High performance Triple Store.
- Fuseki: REST-style SPARQL 1.1 Endpoint access.

These are all part of the Jena Framework.

7. CONCLUSION AND FUTURE WORK

To structure processing chains in a remote sensing services platform, an application ontology has been developed. This ontology organizes the services based on their elementary constituting elements: the operations. These operations are ordered, described and compared within the ontology. Data are also described and linked to the operations.

On the one hand, non-specialized users are able to interact with the RDF graph store, which implements the ontology and its instances, and do not necessary need the help of a knowledge engineer. Creating and editing triples are easily apprehended through the graphical user interface, in a secure, guided and easy way.

On the other hand, knowledge mining in the database is adapted because of the graph structure and its expressiveness. Knowledge base supporting machine learning can support

remote sensing engineers and set rules for quality and consistency analysis during process. Future work will develop a set of axioms to edit new services definitions or optimize those already created.

The use of ontologies is part of a dynamic that is increasingly focused on the pooling of knowledge: Linked Open Data. For this purpose, it is mandatory that the proposed ontology can be used in a larger catalogue application. Integrating the model in a metadata edit and search functions could facilitate many applications and open the possibilities to remote sensing tools to be used by a larger community.

The definitions of data and operations are still generic and improvements can be made. More metadata can be useful to describe the data and enable further research in a large dataset. The integration of the ontology of (Lin et al., 2017) can be an improvement. Many data and operations properties are modelled as Datatype Properties. Switching to Object Properties could lead to new knowledge discovery or possibilities. However, not all the relations can be switched.

Each service is defined by a description following the Dublin Core prescriptions (dc:description). Future work will study the possibility to retrieve the most relevant service from the ontology, based on a query expressed in natural language by non-specialized users in remote sensing.

REFERENCES

Andrés, S., Arvor, D., Mougenot, I., Libourel, T., and Durieux, L. 2017. Ontology-based classification of remote sensing images using spectral rules. *Computers* and *Geosciences*, *102*, pp. 158–166. https://doi.org/10.1016/j.cageo.2017.02.018

Andres, S., Arvor, D., and Pierkot, C. 2012. Towards an Ontological Approach for Classifying Remote Sensing Images (pp. 825–832). IEEE. https://doi.org/10.1109/SITIS.2012.124

Bogorny, V., Engel, P., and Alvares, L. O. 2007. Enhancing the Process of Knowledge Discovery in Geographic Databases Using Geo-Ontologies. In *Data Mining with Ontologies: Implementations, Findings, and Frameworks* (Vol. 45, pp. 160–184). Nigro, Hector Oscar.

Braun, P., Cuzzocrea, A., Leung, C. K., Pazdor, A. G. M., and Tran, K. 2016. Knowledge Discovery from Social Graph Data. *Procedia Computer Science*, *96*, pp. 682–691. https://doi.org/10.1016/j.procs.2016.08.250

Ciccarese, P., and Peroni, S. 2014. The Collections Ontology: Creating and handling collections in OWL 2 DL frameworks. *Semantic Web*, (6), pp. 515–529. https://doi.org/10.3233/SW-130121

Debruyne, C., and O'Sullivan, D. 2016. R2RML-F: Towards Sharing and Executing Domain Logic in R2RML Mappings. In

Proceedings of the Workshop on Linked Data on the Web. Montreal, Canada.

Denaux, R., Dolbear, C., Hart, G., Dimitrova, V., and Cohn, A. G. 2011. Supporting domain experts to construct conceptual ontologies: A holistic approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, *9*, pp. 113–127. https://doi.org/10.1016/j.websem.2011.02.001

Fayyad, U., Piatetsky-Saphiro, G., and Smyth, P. 1996. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, *17*(3), pp. 37–54.

Gómez-Pérez, A., Fernández-López, M., and Corcho, O. 2010. *Ontological engineering: with examples from the areas of knowledge management, e-commerce and the semantic Web*. London; Berlin; Heidelberg [u.a.: Springer.

Grau, B. C., Horrocks, I., Kazakov, Y., and Sattler, U. 2008. Modular Reuse of Ontologies: Theory and Practice. *Journal of Artificial Intelligence Research*, *31*, pp. 273–318. https://doi.org/10.1613/jair.2375

Gruber, T. R. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), pp. 199–220. https://doi.org/10.1006/knac.1993.1008

Han, J., and Kamber, M. 2012. *Data mining: concepts and techniques*. Haryana, India; Burlington, MA: Elsevier.

Hilario, M., Nguyen, P., Do, H., Woznica, A., and Kalousis, A. 2011. Ontology-Based Meta-Mining of Knowledge Discovery Workflows. In N. Jankowski, W. Duch, and K. Grąbczewski (Eds.), *Meta-Learning in Computational Intelligence* (Vol. 358, pp. 273–315). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-20980-2_9

Huang, J., Dou, D., He, L., Hayes, P., and Dang, J. 2010. Ontology-based knowledge discovery and sharing in bioinformatics and medical informatics: A brief survey (pp. 2203–2208).IEEE. https://doi.org/10.1109/FSKD.2010.5569549

Inokuchi, A., Washio, T., and Motoda, H. 2000. An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data. In D. A. Zighed, J. Komorowski, and J. Żytkow (Eds.), *Principles of Data Mining and Knowledge Discovery* (Vol. 1910, pp. 13–23). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45372-5_2

Lin, Y., Xu, H., and Bai, Y. 2017. Semantically Enhanced Catalogue Search Model for Remotely Sensed Imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *10*(4), pp. 1256–1264. https://doi.org/10.1109/JSTARS.2016.2590835

Liu, L., Xue, Y., Guang, J., and Liu, J. 2015. Description of an ontology-based remote sensing model service with an integrated framework environment for remote sensing applications. *Remote Sensing Letters*, *6*(10), pp. 804–813. https://doi.org/10.1080/2150704X.2015.1082207

Miller, H. J., and Han, J. (Eds.). 2009. *Geographic data mining and knowledge discovery* (2nd ed). Boca Raton, FL: CRC Press. Noy, N. F., and Musen, M. A. (2003). The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, *59*(6), pp. 983–1024. https://doi.org/10.1016/j.ijhcs.2003.08.002

Oliva-Santos, R., Maciá-Pérez, F., and Garea-Llano, E. 2014. Ontology-based topological representation of remote-sensing images. *International Journal of Remote Sensing*, *35*(1), pp. 16–28. https://doi.org/10.1080/01431161.2013.858847

Pauwels, P., Terkaj, W., Krijnen, T., and Beetz, J. 2015. Coping with lists in the ifcOWL ontology (pp. 113–122). Presented at the 22nd EG-ICE International Workshop, Eindhoven, Netherlands.

Rowley, J. 2007. The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information Science*, *33*(2), pp. 163–180. https://doi.org/10.1177/0165551506070706

Shen, F., and Lee, Y. 2016. Knowledge Discovery from Biomedical Ontologies in Cross Domains. *PLOS ONE*, *11*(8), e0160005. https://doi.org/10.1371/journal.pone.0160005

Sirin, E., Hendler, J., and Parsia, B. 2003. Semi-automatic Composition of Web Services using Semantic Descriptions. In *Proceedings of the 1st Workshop on Web Services: Modeling, Architecture and Infrastructure* (pp. 17–24). Angers, France: ICEIS Press. Retrieved from http://dblp.uni-trier.de/rec/bibtex/conf/wsmai/SirinHP03

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. 2007. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, *5*(2), pp. 51–53. https://doi.org/10.1016/j.websem.2007.03.004

Srivastava, B., and Koehler, J. 2003. Web Service Composition - Current Solutions and Open Problems. In *Proceedings of ICAPS 2003 Workshop on Planning for Web Services*. Trento, Italy.

Stefanidis, K., Chrysakis, I., and Flouris, G. 2014. On Designing Archiving Policies for Evolving RDF Datasets on the Web. In E. Yu, G. Dobbie, M. Jarke, and S. Purao (Eds.), *Conceptual Modeling* (Vol. 8824, pp. 43–56). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-12206-9_4

Stumme, G., and Maedche, A. 2001. Ontology Merging for Federated Ontologies on the Semantic Web. In *Proceedings of the International Workshop for Foundations of Models for Information Integration* (pp. 413--418).

Tan, H., Adlemo, A., Tarasov, V., and Johansson, M. E. 2017. Evaluation of an Application Ontology. In *CEUR Workshop Proceedings* (Vol. 2050). Bolzano, Italy.

Winston, M., Chaffin, R., and Herrmann, D. 1987. A taxonomy of part-whole relations. *Cognitive Science*, *11*(4), pp. 417–444. https://doi.org/10.1016/S0364-0213(87)80015-0

Yue, P., Di, L., Yang, W., Yu, G., and Zhao, P. 2007. Semantics-based automatic composition of geospatial Web service chains. *Computers* and *Geosciences*, *33*(5), pp. 649–665. https://doi.org/10.1016/j.cageo.2006.09.003