ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume X-4/W4-2024
8th International Conference on Smart Data and Smart Cities (SDSC), 4–7 June 2024, Athens, Greece

# Towards an Automated Transformation of an nD Urban Data Model to a Computational Ontology Network: From UML to OWL, From CityGML 3.0 to "CityOWL"

Diego Vinasco-Alvarez[1], John Samuel[2], Sylvie Servigne[1], Gilles Gesquière[3]

[1] Univ Lyon, INSA Lyon, CNRS, UCBL, LIRIS, UMR5205, F-69621 Villeurbanne, France
(diego.vinasco-alvarez, sylvie.servigne)@insa-lyon.fr
[2] Univ Lyon, CPE Lyon, CNRS, INSA Lyon, UCBL, LIRIS, UMR5205, F-69621 Villeurbanne, France - john.samuel@cpe.fr
[3] Univ Lyon, Univ Lyon 2, CNRS, INSA Lyon, UCBL, LIRIS, UMR5205, F-69676 Bron, France - gilles.gesquiere@univ-lyon2.fr

**Keywords:** Geospatial Semantic Web, Urban Knowledge Graph, 3D City Model, Model-driven Approach, Interoperability, Computational Ontology

## Abstract

As urban data is becoming more complex, researchers are exploring the use of linked data for studying complex urban phenomena. However, mere transformation of heterogeneous data formats to formal knowledge graph formats such as Ontology Web Language (OWL) result in the loss of data. Therefore, we propose an automated approach based on standards to automatically transform conceptual models of urban data to OWL. In this article, we propose reusable configuration rules for transforming UML to OWL as well as the reasoning behind our choice. Our approach is demonstrated by the transformation of a 3D geospatial urban data model, CityGML 3.0, to a network of computational ontologies, informally entitled "CityOWL". We also propose several alignments between the resulting ontologies and existing geospatial, semantic, and temporal linked data standards such as GeoSPARQL, SKOS, and OWL-Time.

## 1. Introduction

3D virtual environments, such as urban digital twins, are increasingly being adopted as tools to gain a better understanding of urban phenomena and city evolution (Barbosa et al., 2014, Batty, 2018). These environments are capable of representing urban information at various scales and levels of detail. Recent years have shown the need to include spatial, temporal and thematic information of city objects to get a holistic view of urban progress. Such urban data is complex to handle and also relies on multiple data standards. For instance, standards like CityGML (Kutzner et al., 2020) can effectively represent 3D geometry, temporal (4D—Time), and thematic information of city objects at different levels of detail, particularly at a city scale. Data that is composed of spatio-temporal 4D data and semantic or thematic data is sometimes categorized as N-Dimensional (nD) data. On the other hand, BIM-IFC can represent detailed building and construction information, including the interiors of a building at a highly granular level.

As the availability of urban data continues to grow, there is an increasing desire to integrate these data (Beck et al., 2021) to obtain a comprehensive view of urban objects from various sources of information. However, achieving such integration poses challenges due to the need to understand multiple heterogeneous data standards based on different data models and formats. Moreover, these urban data models are rapidly evolving.

Furthermore, gaps in interoperability between heterogeneous (urban) data standards make reusing these standards difficult, with various types of heterogeneity existing, such as semantic, structural/schematic, and data models (Kutzner, 2016).

Additionally, data transformation between different data standards may pose the risk of data loss, limiting the reusability of the transformed data. The loss of semantic data serves as a barrier to the adoption of urban digital twin applications (Lei et al., 2023). Data transformations that take into account the underlying data models and formats can help mitigate data loss (Bohring and Auer, 2005).

In this article, we focus on the following two research questions.

- **RQ1**: How can nD urban data integration approaches ensure that standardized urban data can be easily reused, even as data standards evolve?
- **RQ2**: How do data model transformation rules effect the conciseness of the targeted data model?

The guiding principle of this work is to present a generically applicable standards-based model-driven approach for transforming conceptual nD urban data models (Figure 1). This will promote the reusability and interoperability of our resulting contributions. In turn, the implementations presented in this article will be reproducible.

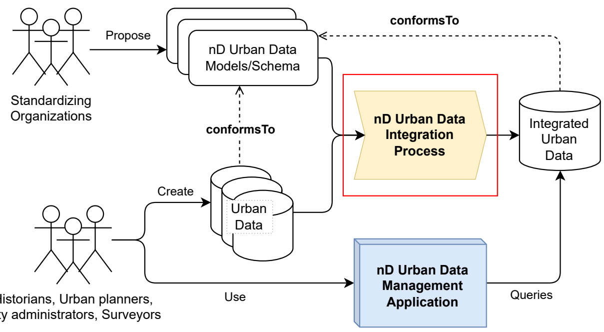


Figure 1. Goal: Approach for integrating nD urban data.

The primary contribution of our work is the utilization of a model-driven approach for transforming nD Urban Data Models to computational ontologies. We provide detailed insights into our use of international standards. We also provide insights and reasoning behind the choice of transformation rules for our approach. The application of our approach is demonstrated through its implementation using existing and evolving urban data models, i.e., CityGML 2.0 and 3.0 (Kutzner et al., 2020) and geospatial data models. Additionally, we highlight the consideration given to extensions of these models in our approach.

The article discusses the state of the art in Section 2, including its limitations. Section 3 outlines the development of the proposed approach. The experiments and results of this approach are detailed in Sections 4 and 5 respectively. Pros and cons of our proposed approach are deliberated in Section 6. Lastly, Section 7 concludes the article and outlines the future course of actions.

## 2. State of the Art

Much of the nD urban data used to create urban digital twin applications come from the domains of Geospatial Information Systems (GIS) and Building Information Modeling (BIM). Data integration approaches in these information domains have been examined in previous research (Beck et al., 2021), particularly in terms of conversion, extension, (inter)-linking, and merging (Figure 2). Regarding data models, **conversion** entails the transformation or *translation* of a source data model to a target data model resulting in the creation of new data model. Similarly, data can be transformed to conform from a source data model to a target data model. In the case of **extension**, an existing model is expanded to facilitate lossless conversion. **(Inter)-linking** involves materializing explicit links between the concepts and relationships of data models to support integration as needed. Lastly, **merging** encompasses the consolidation of multiple models to create a new merged model.

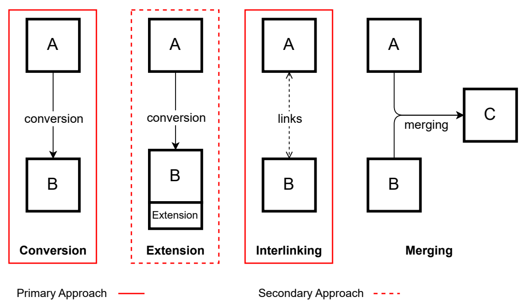


Figure 2. 4 types of urban data integration approaches (Beck et al., 2021) in the context of Building Information Modeling (BIM) and Geographic Information Systems (GIS) information integration. These approaches are applied to two data models *A* and *B*.

This article focuses on the automated integration of models thro-ugh **conversion** and **extension** to facilitate **linking**, as illustrated in Figure 2. We consider the underlying conceptual models and identify UML and XML Schema (XSD) as source languages, with OWL 2 DL (computational ontology) as the target language. Computational ontologies (Studer et al., 1998, Uschold and Gruninger, 2004) are machine-readable, highly formal, and support rich rules.

### 2.1 Conversion approaches

To achieve this objective, we took into consideration previous works that convert urban data models and schema to ontological and knowledge graph formats (utilizing RDF, OWL, etc.) through automated transformation. Works such as (Kramer et al., 2015, Kyzirakos et al., 2018, Usmani et al., 2020, Zedlitz and Luttenberger, 2012) propose transformations from XML Schema to ontologies. Some existing works also suggest the transformation of UML models to ontologies (Gasevic et al., 2004, De Paepe et al., 2017). Some of these approaches follow the UML to OWL transformation rules proposed in ISO 19150-2 (ISO/TC 211, 2021) including international data interoperability and standardization initiatives and organizations such as the INSPIRE program and the Open Geospatial Consortium (ARE3NA project, 2017, Echterhoff, 2017). (Jetlund et al., 2019) proposes a UML to OWL transformation approach that takes into consideration the proposed transformations in (ARE3NA project, 2017, Echterhoff, 2017).

(Jetlund et al., 2019, Echterhoff, 2017) note that several approaches exist for transforming the geospatial concepts from the General Feature Model (GFM)[1] to OWL using the rules defined in ISO 19150-2. The GFM is implemented as a UML profile proposed in ISO 19103 (ISO/TC 211, 2015a) and ISO 19109 (ISO/TC 211, 2015b). (Jetlund et al., 2019, Echterhoff, 2017) also note that the UML to OWL transformations rules proposed in ISO 19150-2 contain several limitations or ambiguities that imply interpreting or adapting the standard. For example, `<<Enumeration>>` (used to define extensible and reusable data types for categorizing geospatial features) could be represented using extensible structured vocabulary standards such as the Simple Knowledge Organisation System (SKOS) standard. Additionally, the transformation of `<<Union>>` (used to abstractly group sets of UML attributes) is potentially insufficient.

A comparison of these approaches is provided in Table 1. As illustrated in Figure 1, our aim is to transform the data models using UML models or XML schema to later use these models to support the transformation of geospatial data instances. For the purposes of this goal, the approaches proposed in (Jetlund et al., 2019) is suitable as it is configurable and reuses the existing nD vocabularies proposed by the ISO 191xx series of standards. These vocabularies provide extensible abstract classes for geometry, geospatial concepts, and time that are used in international geospatial data standards such as CityGML.

### 2.2 Linking approaches

In our approach, we use a linking approach to support multiple geospatial data standards. Linking approaches declare correspondences (or relations) between entities defined in the data model or between data instances. In computational ontologies, these entities often include the classes, properties, or individuals from two ontologies (Euzenat and Shvaiko, 2013). A set of correspondences between two ontologies is often called an alignment (Figure 3). Alignments are typically declared pairwise between a set of ontologies in order to define an ontology network. There currently exist several automatic and manual ontology alignment approaches for nD urban data models and data (Usmani et al., 2021, Vilgertshofer et al., 2017).

### 2.3 Synthesis

Knowledge graph formats and computational ontologies have been shown to provide flexible methods for representing urban

[1] According to Model Driven Architecture (MDA), metamodels are composed of the languages used to define data models (Atkinson and Kühne, 2003)

| Approach | Source model language | Vocabulary reuse? | Configurable? |
|---|---|---|---|
| (Jetlund et al., 2019) | UML | Yes | Yes |
| (De Paepe et al., 2017) | UML | No | Yes |
| (Zedlitz and Luttenberger, 2012) | UML | Yes | No |
| (Kyzirakos et al., 2018) | XML Schema, JSON Schema, DBMS | Yes | No |
| (Kramer et al., 2015) | XML Schema | No | No |
| (Usmani et al., 2020) | XML Schema | No | No |

Table 1. A comparison of previous works in model-driven transformations towards OWL computational ontologies. Some of these approaches reuse existing geospatial, temporal, and/or semantic vocabularies. In particular, some UML to OWL transformation approaches provide mechanisms for configuring their transformations.

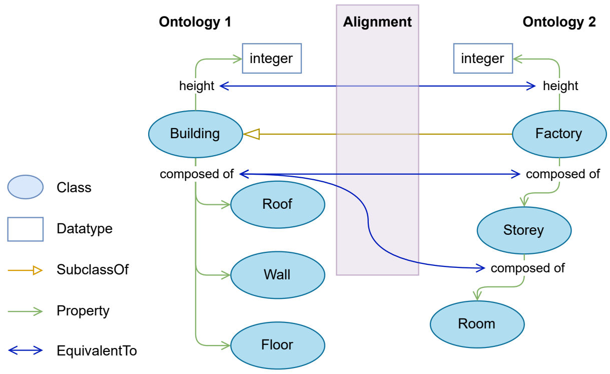


Figure 3. Model and data linking approaches.

information and facilitate data integration through linking. As progress is made towards improving model-driven transformations between abstract and physical data models towards computation ontologies, the easier it becomes to integrate urban information under common data description languages and formats. Approaches such as (Jetlund et al., 2019) rely on standardized transformation rules that can be easily reused and adapted to different use-cases.

The work presented in this article proposes several additions to the UML to OWL transformation rules identified in these works to improve the model-driven geospatial UML to OWL transformations of concepts such as Unions and CodeLists. The proposed rules focus on limiting the loss of semantic information while improving the conciseness of the transformation results. These rules are detailed in the following section.

## 3. Transformation Methodology

This work provides an approach based on the ISO 19150-2 UML to OWL transformation mappings. As these mappings are generic, they can theoretically be applied to any existing UML model (including models that take into consideration the GFM metamodel and other geospatial data modeling standards from ISO 191xx). As noted in the previous section, the known best-practices, ambiguities, and limitations of using ISO 19150-2 must be considered during its implementation. Building off of the works denoted in the previous section, we propose a strategy for transforming the `<<Union>>`, `<<CodeList>>`, and `<<Enumeration>>` concepts from the GFM metamodel to OWL.

### 3.1 Transforming Unions

(Echterhoff, 2017, Jetlund et al., 2019) note that although Unions exist in both UML and OWL, their use is not identical and a direct transformation between `<<Union>>` and `owl:Union` is not straightforward. According to ISO 19150-2, the attribute members of a `<<Class>>` are transformed into `owl:ObjectProperties` or `owl:DatatypeProperties` depending on if the classifier of the attribute is a primitive type or another `<<Class>>`. UML models using the GFM can declare a `<<Union>>` to group a set of attributes. The following transformation strategy is proposed depending on what the attribute members of the `<<Union>>` would be transformed into. There are three possibilities:

**Case 1:** If the attribute members of the `<<Union>>` would be transformed into `owl:ObjectProperties`, transform the `<<Union>>` into an `owl:ObjectProperty`. Declare the `owl:ObjectProperties` created from the attribute members of the `<<Union>>` as `rdfs:subPropertyOf` the object property.

**Case 2:** If the attribute members of the `<<Union>>` would be transformed into `owl:DatatypeProperties`, transform the `<<Union>>` into a `owl:DatatypeProperty`. Declare the `owl:DatatypeProperties` created from the attribute members of the `<<Union>>` as `rdfs:subPropertyOf` the datatype property.

**Case 3:** If the attribute members of the `<<Union>>` would be transformed into a combination of both `owl:ObjectProperties` and `owl:DatatypeProperties`, "flatten" any associations or attributes targeting the `<<Union>>` and the attribute members of the `<<Union>>` into `owl:ObjectProperties` and `owl:DatatypeProperties` as proposed in (Echterhoff, 2017).

The first two cases are inspired from (Zedlitz and Luttenberger, 2012) who propose transformations of `<<Union>>` to `owl:Class` and `<<Union>>` attributes to properties and subproperties (figure 4). However, we do not propose transforming the original `<<Union>>` into a `owl:Class`. (Echterhoff, 2017) notes that transforming `<<Union>>` in this way may result in redundant "intermediate" individuals being generated when instantiating the class generated from the union. Instead, the union should represent a parent property of the attribute members it groups as illustrated in Figure 4.

Because `owl:ObjectProperties` and `owl:DatatypeProperties` cannot share the same superproperty, cases 1 and 2 cannot be used in conjunction. A "flattening" approach (Echterhoff, 2017) can be used as an alternative in this case (Figure 5). This approach is less desirable to mappings 1 and 2, as it implies the semantic loss of the `<<Union>>` itself during transformation. As a trade-off, less structural heterogeneity is created between the original model and generated ontology by producing a more concise representation of a `<<Union>>`.

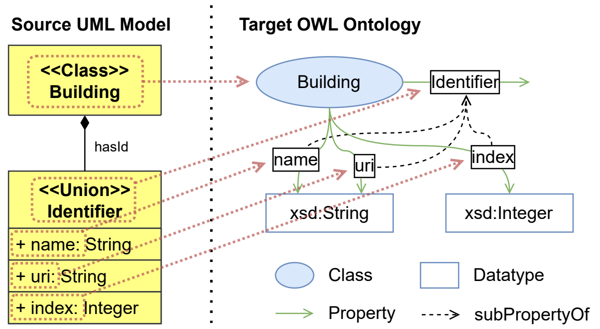


Figure 4. An example of a Union transformation according to case 2. The union, *Identifier*, contains 3 attributes with primitive datatype classifiers that would be transformed into datatype properties. These properties are declared as a subproperties of the new *Identifier* property.

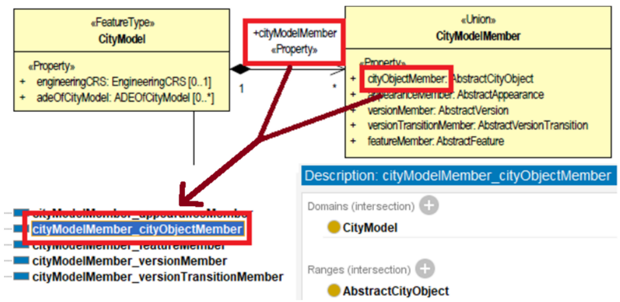


Figure 5. Flattening of the union, *core:CityModelMember* by combining its attributes with the composition, *core:cityModelMember*, (top) and their transformation to OWL (bottom) from the CityGML 3.0 UML model (image from (Vinasco-Alvarez et al., 2021)).

### 3.2 Transforming CodeList and Enumeration

According to (ISO/TC 211, 2021), a `<<CodeList>>` in OWL can be transformed into an internal or external SKOS concept scheme. `<<Enumeration>>` can be transformed to either `owl:DataUnionOf` according to (ISO/TC 211, 2021) or to SKOS concept schemes as there is little distinction between a `<<CodeList>>` and a `<<Enumeration>>` (besides the requirement that a `<<CodeList>>` must be extensible) (Echterhoff, 2017, ARE3NA project, 2017). Thus, we propose transforming `<<Enumeration>>` to `<<CodeList>>` and transforming `<<CodeList>>` to externally defined SKOS concept schemes based on the approach proposed in (Echterhoff, 2017). Declaring the schemes externally allows for their vocabularies to be managed and versioned independently of the larger data model and provides a mechanism for extending their vocabularies. Unlike `<<CodeList>>`, `<<Enumeration>>` propose possible values for their instantiation. The original values of the `<<Enumeration>>` must be generated as a part of the SKOS concept scheme.

The following section details how we implemented and evaluated these propositions with a standardized nD urban data model.

## 4. Experiments

In order to evaluate the proposed transformation rules, the CityGML 3.0 conceptual model is used as a case study. As mentioned before, CityGML standard and their extensions can represent geometry, semantic, thematic, and temporal information. In particular, the model is composed of 17 modules, including 11 thematic modules for describing information from several domains of the urban environment such as construction, vegetation, transportation, etc. The conceptual model of CityGML is available as a UML model[2].

Figure 6 illustrates the different packages of the CityGML conceptual model and their dependencies to the GFM model and other ISO 191xx standards. This figure organizes the relevant models and metamodels (languages) according to the metamodeling levels (M1 and M2 respectively) of the Object Management Group's metamodeling infrastructure (Object Management Group, 2003). This figure also illustrates the target computational ontologies to be produced by applying the aforementioned transformation rules. Dependencies between the OWL ontologies represent the proposed alignments between these ontologies, creating a network of ontologies.

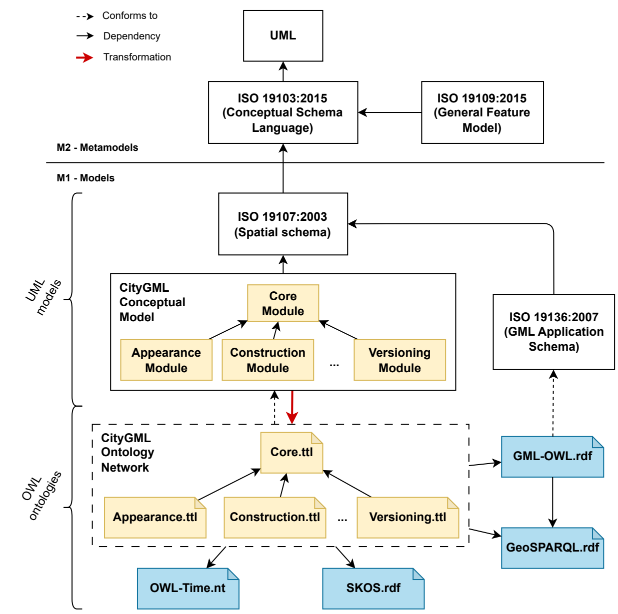


Figure 6. Dependencies and conformance between the CityGML, GML, and ISO 191xx models. The CityGML model is represented before and after transformation. Among the OWL ontologies, pre-existing ontologies that are being reused are highlighted in blue.

The proposed alignments are declared in UML before transformation whenever possible. The majority of our correspondences are specializations of classes declared with `rdfs:subClassOf`. Additionally, Table 2 lists the "non-standard" ontology prefixes used in this article. The following subsections detail our proposed alignments.

### 4.1 Geospatial and Geometric Alignments

While the geometry of CityGML 3.0 data is encoded according to the GML 3.2 standard, the CityGML conceptual model represents geometry through the abstract spatial classes of the ISO 19107 standard (ISO/TC 211, 2019). As noted in (Jetlund et al., 2019), this provides two approaches for representing geometry.

[2] `https://github.com/opengeospatial/CityGML-3.0CM/releases/download/3.0.0-final.2021.02.23/XMI.Files.zip`

 

| Prefix | URI |
|---|---|
| skos | http://www.w3.org/2004/02/skos/core# |
| geo | http://www.opengis.net/ont/geosparql# |
| gml | http://www.opengis.net/ont/gml |
| time | http://www.w3.org/2006/time# |
| core | https://dataset-dl.liris.cnrs.fr/rdf-owl-urban-data-ontologies/Ontologies/CityGML/3.0/core# |
| vers | https://dataset-dl.liris.cnrs.fr/rdf-owl-urban-data-ontologies/Ontologies/CityGML/3.0/versioning# |
| wksp | https://dataset-dl.liris.cnrs.fr/rdf-owl-urban-data-ontologies/Ontologies/Workspace/3.0/workspace# |

Table 2. The prefixes used in this article with their respective URIs.

The first implies an alignment with the official ontological representation of GML 3.2[3]. The second implies an alignment with the official ontological representation of the ISO 19107 standard[4].

This work utilizes the former approach the GML 3.2 ontology is more concise than the ISO 19107 ontologies. The GML 3.2 ontology also aligns with the GeoSPARQL standard, a widely used standard for representing geospatial information. Figure 7 illustrates the proposed alignments between the CityGML Core module and the GML and GeoSPARQL ontologies (above). The correspondences between CityGML and GML are mostly composed of references between `core:AbstractSpace` and the primitive geometry classes of GML. Because these properties are functionally similar to the more general, `geo:hasGeometry`, they are declared as subproperties of this property.

### 4.2 Temporal and Semantic Alignments

The OWL-Time and SKOS ontologies are used to represent several temporal and semantic concepts from the CityGML model.

In CityGML, the superclass `core:AbstractFeatureWithLifespan` represents any feature with time dependent properties. 4 properties of this class are used to represent two time intervals of any city feature. `core:creationDate` and `core:terminationDate` denote when the feature was created and destroyed in the real-world. `core:validFrom` and `core:validTo` denote when the feature was added and removed from the CityGML dataset. As shown in Figure 7 (below), these properties are declared as `rdfs:subPropertyOf` the OWL-Time `time:Instant` class.

Aligning CityGML with OWL-Time in this way permits inferring temporal relations between geospatial city features (Batsakis et al., 2017). For example, given a building, $A$, constructed after the deconstruction of some building, $B$, one could infer that $A$ exists temporally after $B$.

Concerning the SKOS ontology, concept schemes are used to represent CodeList values and certain Enumeration values (as proposed in section 3.2). Figure 7 (below) illustrates how the `vers:TransactionType` enumeration from the CityGML Versioning module can be represented in this way. This enumeration denotes the type of change a city object undergoes between two versions of the city at two different states: `insert` reflects a city object being added to the city; `delete` reflects a city object being removed; `replace` reflects a city object being updated or modified between two versions. This concept scheme can be easily extended or replaced with other vocabularies for categorizing city object changes such as the vocabularies proposed by (Renolen, 2000).

### 4.3 CityGML Extensions

Extensions to the CityGML model are supported through Application Domain Extensions (ADEs). To explore how ADEs can be represented in OWL, the Workspace conceptual model proposed in (Samuel et al., 2020) is remodeled as a CityGML 3.0 ADE. This ADE uses the CityGML 3.0 Versioning module by providing concepts and data structures for representing concurrent scenarios of urban evolution (such as `wksp:PropositionSpace` and `wksp:ConcensusSpace`). As detailed in the CityGML 3.0 conceptual model documentation, this ADE is formalized as a UML package that imports classes from CityGML. The results of this transformation are illustrated in Figure 7.

## 5. Results

To effectuate transformation, the ShapeChange[5] transformation tool is used. This tool can be configured to implement many of the transformation rules proposed in (Jetlund et al., 2019, Echterhoff, 2017, ARE3NA project, 2017). Development of the Workspace ADE is effectuated in Enterprise Architect[6], a tool for creating and maintaining UML models. A post-processing step is used to correct logical inconsistencies produced by Shape-Change (such as producing datatype properties with classes in their domain). This step ensures that the generated ontologies fall under the constraints for the description logic dialect of OWL, OWL-DL.

We refer to the generated ontology network as "CityOWL" aligning with the naming conventions of CityGML and its JSON encoding, CityJSON[7]. Using the transformations discussed in this article two CityOWL representations are produced. One is a more constrained "Closed World Assumption" (CWA) representation which features class and property constraints such as, universal and existential restrictions, cardinality, and declarations of property `rdfs:domain` and `rdfs:range`. This network of ontologies falls under the description logic expressivity, $SHQ^{(D)}$. A more concise "Open World Assumption" (OWA) representation of CityGML is also produced that only contains the classes and properties generated from the model with no restrictions. This network of ontologies falls under the description logic expressivity, $ALH^{(D)}$. Table 3 provides a comparison of the number of logical axioms, classes, properties, and datatypes each ontology network contains.

## 6. Discussion

We ensured the aforementioned contributions are available in reproducible, interoperable, and accessible manners. To do this, reproducible transformation workflows were created through

[3] https://schemas.opengis.net/gml/3.2.1/gml_32_geometries.rdf
[4] https://def.isotc211.org/ontologies/iso19107/
[5] https://shapechange.net/
[6] https://sparxsystems.com/products/ea/index.html
[7] https://cityjson.org/

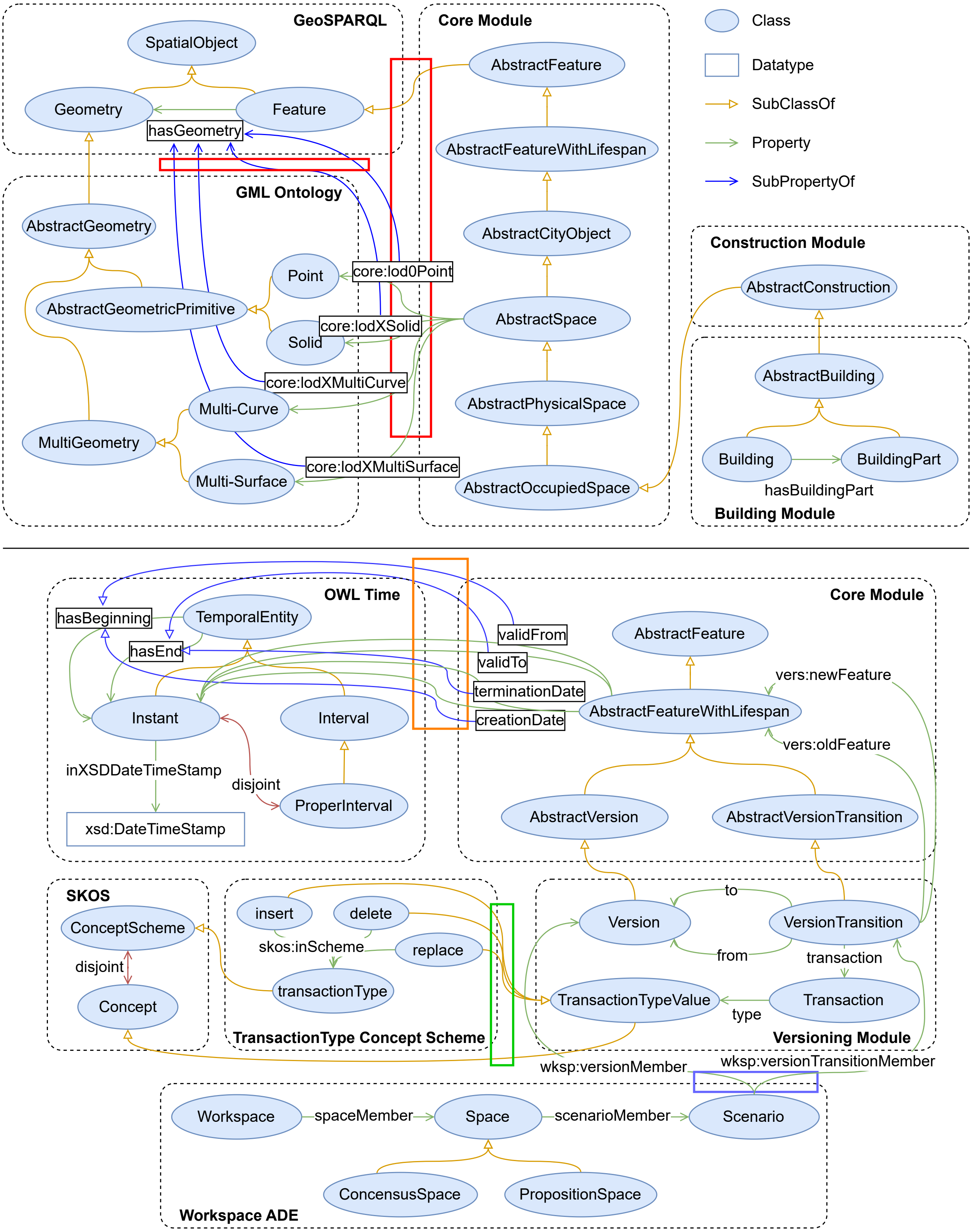


Figure 7. A subset of the geospatial and geospatial classes of GML and GeoSPARQL with several abstract classes and properties of the generated CityGML ontology network and their correspondences (above). Also, a subset of the classes and properties of OWL-Time, SKOS, the generated CityGML ontology network, and the proposed Workspace ADE (below). Proposed alignments between CityGML and GeoSPARQL/GML are highlighted in red. Proposed alignments between CityGML and OWL-Time are highlighted in orange. Proposed alignments between CityGML and SKOS are highlighted in green. Proposed alignments between CityGML and the Workspace ADE are highlighted in blue.

| Ontology | Axioms | Classes | Properties | Datatypes |
|---|---|---|---|---|
| OWA CityOWL | 4193 | 425 | 568 | 13 |
| CWA CityOWL | 5041 | 433 | 559 | 13 |

Table 3. A comparison of the CityOWL ontology networks generated from the CityGML 3.0 conceptual model.

the use of Python, Shell scripts, and Docker[8] container technology. Table 4 provides links and identifiers for accessing these workflows and the other experimentation results of this work. Documentation, user guides, installation instructions, and documented of technical issues are provided with each of these contributions whenever applicable. We provide links to these contributions through software archiving services such as Software Heritage[9] (table 4). To illustrate how data models and data integrated using this approach could be used, element 9 of table 4 links to a proof of concept 3D urban web application that uses these integrated data to contextualize a 3D city scene. In addition, element 10 provides a link to the alignments proposed in figure 7. All code, documentation, data models, and data are produced using a *Free and Open Software* (FOSS) approach to providing readily available and easily extensible software to international communities (Wheeler, 2007). Therefore, we use an open LGPL-2 license[10] for licensing the aforementioned contributions.

Furthermore, the following approach was taken to improve the interoperability of the produced ontological data models following the principle of reusing data standards. After transformation, linking and extension are implemented to reuse existing standards. In the case where an existing data model in the targeted modeling language and data format is available that *completely* meets the identified needs of our integration use-case, that data model is **linked** through alignment. This is proposed since a linking approach does not modify the target data model being integrated, maximizing interoperability. In the case where an existing data model is available that *partially* meets the identified needs of a particular use-case or application, that standard is **extended** and **linked**. Linking and extension are preferred over a merging approach (as introduced in section 2) as a merging approach creates a new, tertiary data model instead of reusing existing models.

Finally, the proposed rules for creating computational ontologies from the CityGML 3.0 conceptual model were proposed for potential applications. The use of other rules may be more applicable for other use-cases. For example, approaches that require testing if data instances conform to the classes and properties of the generated ontology may prefer creating a more constrained CWA ontology. This permits data validation through class and property restrictions such as cardinality and existential quantification. Other applications may just require a set of classes and properties to provide interoperability. These use-cases can likely generate a less constrained OWA ontology.

## 7. Conclusion and Future Works

In this article, we proposed an automated integration approach based on standardized rules for transforming UML to OWL. These rules also take into consideration the transformation of geospatial concepts from the GFM metamodel. This approach was applied to the conceptual model of CityGML 3.0 to create a network of computational ontologies. These ontologies remain interoperable with the original CityGML standard and several existing spatio-temporal ontologies. A large effort was made to ensure these rules are reproducible through configurable transformation tools such as ShapeChange and UD-Graph.

As part of future works, we want to integrate more complex rules for handling the evolution of cities or urban data. This will require the use of additional standards, especially those associated with document corpuses and their metadata. Furthermore, we also wish to explore the querying of 3D geometry along with other urban data. Currently, native 3D geospatial queries are not supported by standards such as GeoSPARQL, however, this functionality is in development by standardizing organizations such as the Open Geospatial Consortium[11].

**Declaration of competing interests and Acknowledgements**

The authors declare that they have no conflict of interest. The authors would like to thank University Lumière Lyon 2 and INSA Lyon for funding the PhD Thesis of Diego Vinasco-Alvarez. This work has been done within the LIRIS VCity project[12].

## References

ARE3NA project, 2017. Guidelines for the RDF encoding of spatial data.

Atkinson, C., Kühne, T., 2003. Model-driven development: a metamodeling foundation. IEEE Software 20(5), 36-41. *Software, IEEE*, 20, 36 - 41.

Barbosa, L., Pham, K., Silva, C., Vieira, M. R., Freire, J., 2014. Structured Open Urban Data: Understanding the Landscape. *Big Data*, 2(3), 144–154. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4174913/.

Batsakis, S., Petrakis, E. G. M., Tachmazidis, I., Antoniou, G., 2017. Temporal representation and reasoning in OWL 2. *Semantic Web*, 8(6), 981–1000. https://www.semantic-web-journal.net/system/files/swj855.pdf. Publisher: IOS Press.

Batty, M., 2018. Digital twins. *Environment and Planning B: Urban Analytics and City Science*, 45(5), 817–820. https://doi.org/10.1177/2399808318796416. Publisher: SAGE Publications Ltd STM.

Beck, S. F., Abualdenien, J., Hijazi, I. H., Borrmann, A., Kolbe, T. H., 2021. Analyzing Contextual Linking of Heterogeneous Information Models from the Domains BIM and UIM. *ISPRS International Journal of Geo-Information*, 10(12), 807. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute.

Bohring, H., Auer, S., 2005. Mapping XML to OWL ontologies. *Marktplatz Internet: Von e-Learning bis e-Payment, 13. Leipziger Informatik-Tage (LIT 2005)*, Gesellschaft für Informatik e. V., 147–156.

[8] `https://www.docker.com/`
[9] `https://www.softwareheritage.org/`
[10] `https://www.gnu.org/licenses/lgpl-3.0.html`
[11] `https://www.ogc.org/`
[12] `https://projet.liris.cnrs.fr/vcity/`

| | Technical contribution | Software Heritage ID |
|---|---|---|
| 1. | CityOWL Ontologies | swh:1:dir:8f07f77f3ed973f04919cc4af3b2762d20375b95 |
| 2. | UD-Reproducibility | swh:1:dir:4d1ac3f824c1ffa3c0d93a8b946e007e716195f1 |
| 3. | Source UML Models | swh:1:dir:5bc538c2fabb300cf3cf6c4750aecb1e1471d9e2 |
| 4. | CityOWL CWA results | swh:1:dir:ac046f30f5841ea2bcaee4d61e11ac638b8be835 |
| 5. | CityOWL OWA results | swh:1:dir:924d235c365377ff51b30d1149702435f33305cb |
| 6. | ShapeChange configuration files | swh:1:dir:28316a8fcb9f4d47ca4b7e4217af86d0793dcbd8 |
| 7. | UD-Graph (transformation component) | swh:1:dir:2fce25376177276e02d853250fba33db83f2c4a0 |
| 8. | Proposed SKOS Concept Schema | swh:1:cnt:62deff5243eb92b8807099b81ac84cf27ed9a0fb;path=/ |
| 9. | Demonstrative 3D Web Application | `http://demo.liris.cnrs.fr/vcity/ud-demo-graph-static-gratteciel/` |
| 10. | Proposed alignments | swh:1:cnt:b11daffa4648a5100d8d8517f426014239fc7456 |

Table 4. Software Heritage identifiers of transformation tools, configuration files, data models, and datasets discussed in this dissertation. Software Heritage IDs can be used at `https://archive.softwareheritage.org/` to view the archived resources.

De Paepe, D., Thijs, G., Buyle, R., Verborgh, R., Mannens, E., 2017. Automated UML-Based Ontology Generation in OSLO2. 93–97.

Echterhoff, J., 2017. Testbed-12 ShapeChange Engineering Report. Public Engineering Report 16-020, Open Geospatial Consortium.

Euzenat, J., Shvaiko, P., 2013. *The Matching Problem*. 2nd ed. 2013 edn, Springer Berlin Heidelberg : Imprint: Springer.

Gasevic, D., Djuric, D., Devedzic, V., Damjanovi, V., 2004. Converting UML to OWL ontologies. *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, WWW Alt. '04, Association for Computing Machinery, New York, NY, USA, 488–489.

ISO/TC 211, 2015a. ISO 19103:2015 - Geographic information — Conceptual schema language.

ISO/TC 211, 2015b. ISO 19109:2015, Geographic information - Rules for application schema.

ISO/TC 211, 2019. ISO 19107:2019(en), Geographic information — Spatial schema.

ISO/TC 211, 2021. ISO 19150-2:2015(en), Geographic information — Ontology — Part 2: Rules for developing ontologies in the Web Ontology Language (OWL).

Jetlund, K., Onstein, E., Huang, L., 2019. Adapted Rules for UML Modelling of Geospatial Information for Model-Driven Implementation as OWL Ontologies. *ISPRS International Journal of Geo-Information*, 8(9). https://www.mdpi.com/2220-9964/8/9/365.

Kramer, T. R., Marks, B. H., Schlenoff, C. I., Balakirsky, S. B., Kootbally, Z., Pietromartire, A., 2015. Software Tools for XML to OWL Translation. Last Modified: 2018-11-10T10:11-05:00.

Kutzner, T., 2016. Geospatial data modelling and model-driven transformation of geospatial data based on uml profiles. PhD Thesis, Technische Universität München.

Kutzner, T., Chaturvedi, K., Kolbe, T. H., 2020. CityGML 3.0: New Functions Open Up New Applications. *PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88(1), 43–61. http://link.springer.com/10.1007/s41064-020-00095-z.

Kyzirakos, K., Savva, D., Vlachopoulos, I., Vasileiou, A., Karalis, N., Koubarakis, M., Manegold, S., 2018. GeoTriples: Transforming geospatial data into RDF graphs using R2RML and RML mappings. *Journal of Web Semantics*, 52-53, 16–32.

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.

Object Management Group, 2003. UML 2.0 Infrastructure Specification.

Renolen, A., 2000. Modelling the Real World: Conceptual Modelling in Spatiotemporal Information System Design. *Transactions in GIS*, 4(1), 23–42. https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-9671.00036.

Samuel, J., Servigne, S., Gesquière, G., 2020. Representation of concurrent points of view of urban changes for city models. *Journal of Geographical Systems*, 22(3), 335–359. https://doi.org/10.1007/s10109-020-00319-1.

Studer, R., Benjamins, V. R., Fensel, D., 1998. Knowledge engineering: principles and methods. Data Knowl Eng 25(1-2):161-197. *Data & Knowledge Engineering*, 25, 161-197.

Uschold, M., Gruninger, M., 2004. Ontologies and semantics for seamless connectivity. *ACM SIGMod Record*, 33(4), 58–64.

Usmani, A. U., Jadidi, M., Sohn, G., 2020. Automatic Ontology Generation of BIM and GIS Data. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B4-2020, Copernicus GmbH, 77–80. ISSN: 1682-1750.

Usmani, A. U., Jadidi, M., Sohn, G., 2021. Towards the Automatic Ontology Generation and Alignment of BIM and GIS Data Formats. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, VIII-4-W2-2021, Copernicus GmbH, 183–188. ISSN: 2194-9042.

Vilgertshofer, S., Amann, J., Willenborg, B., Borrmann, A., Kolbe, T. H., 2017. Linking BIM and GIS Models in Infrastructure by Example of IFC and CityGML. 133–140. Publisher: American Society of Civil Engineers.

Vinasco-Alvarez, D., Samuel, J., Servigne, S., Gesquière, G., 2021. Towards Limiting Semantic Data Loss in 4D Urban Data Semantic Graph Generation. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, VIII-4/W2-2021, 37–44. https://isprs-annals.copernicus.org/articles/VIII-4-W2-2021/37/2021/.

Wheeler, D. A., 2007. Why open source software/free software (oss/fs, floss, or foss)? look at the numbers.

Zedlitz, J., Luttenberger, N., 2012. Transforming Between UML Conceptual Models and OWL 2 Ontologies. *Terra Cognita 2012 Workshop*.