TOWARDS LIMITING SEMANTIC DATA LOSS IN 4D URBAN DATA SEMANTIC GRAPH GENERATION

Diego Vinasco-Alvarez[a], John Samuel[b], Sylvie Servigne[c], Gilles Gesquière[a]

[a]Université Lumière Lyon 2, LIRIS UMR-CNRS 5205, Université de Lyon, Lyon, France -
(diego.vinasco-alvarez, gilles.gesquiere)@univ-lyon2.fr;
[b]CPE Lyon, LIRIS, UMR-CNRS 5205, Université de Lyon, Lyon, France - john.samuel@cpe.fr;
[c]INSA Lyon, LIRIS UMR-CNRS 5205, Université de Lyon, Lyon, France - sylvie.servigne@insa-lyon.fr

ABSTRACT:

To enrich urban digital twins and better understand city evolution, the integration of heterogeneous, spatio-temporal data has become a large area of research in the enrichment of 3D and 4D (3D + Time) semantic city models. These models, which can represent the 3D geospatial data of a city and their evolving semantic relations, may require data-driven integration approaches to provide temporal and concurrent views of the urban landscape. However, data integration often requires the transformation or conversion of data into a single shared data format, which can be prone to semantic data loss. To combat this, this paper proposes a model-centric ontology-based data integration approach towards limiting semantic data loss in 4D semantic urban data transformations to semantic graph formats. By integrating the underlying conceptual models of urban data standards, a unified spatio-temporal data model can be created as a network of ontologies. Transformation tools can use this model to map datasets to interoperable semantic graph formats of 4D city models. This paper will firstly illustrate how this approach facilitates the integration of rich 3D geospatial, spatio-temporal urban data and semantic web standards with a focus on limiting semantic data loss. Secondly, this paper will demonstrate how semantic graphs based on these models can be implemented for spatial and temporal queries toward 4D semantic city model enrichment.

1. INTRODUCTION

The continuing amelioration of semantic 3D city models has provided powerful tools for comprehending the dynamics of the constantly evolving urban landscape. Data-driven approaches such as the construction of virtual environments such as digital-twins (Batty, 2018, Julin et al., 2018, Schrotter and Hürzeler, 2020) can help visualize and simulate city events and dynamics both real and imaginary, over time (Biljecki et al., 2015, Chaturvedi and Kolbe, 2019, Jaillot et al., 2020, Samuel et al., 2020).

Currently, the Open Geospatial Consortium's (OGC)[1] CityGML[2] open data model is often used to facilitate the storage and exchange of virtual 3D city models. In this model, all city objects can be represented in up to five different, well-defined levels-of-detail (LOD0 to LOD4) with increasing accuracy but structural complexity (e.g., a building is composed of walls, roofs, etc.) (Kolbe, 2009). It defines the three-dimensional geometry, topology, semantics, and appearance of the most relevant topographic objects in the urban context (Gröger and Plümer, 2012).

The CityGML standard is still evolving (Kutzner et al., 2020) to represent all the complex data requirements of 4D (3D + Time) semantic city models (Chaturvedi and Kolbe, 2019) and may need enrichment through data integration of other heterogeneous data sources to achieve more complete or detailed views of the urban landscape. To facilitate integration, the use of ontologies has been promoted as they provide flexible, machine-processable formalizations of data models as semantic graphs

(Claramunt, 2020, Psyllidis, 2015). Ontologies allow then to create semantic urban graphs based on urban data which may require transformation or conversion of data towards a single common data format. However, transformation approaches are naturally prone to semantic data loss and thus limiting this data loss poses a principal integration challenge.

Another challenge – within the context of multisource temporal urban data integration – is the inconsistent use of identifiers of city objects (Chaturvedi et al., 2017). Different snapshots of a city from different sources may use different identifiers for the same city objects between concurrent and successive representations of that city. Recently, graph-based change detection approaches (Jaillot et al., 2020, Nguyen and Kolbe, 2020) have been proposed to calculate which city objects are being referenced across different city snapshots, even when there are geometric, or semantic differences between the same object in different temporal representations of the city.

As ontologies and graph formats have proven beneficial towards the semantic enrichment of interoperable multidimensional city models, this paper proposes a model-centric ontology-based data integration approach towards limiting semantic data loss in 4D semantic urban data transformations to semantic graph formats. This contribution consists of several aspects:

1. A proposed 4D semantic urban data model, based on a network of ontologies (Euzenat, 2014), incorporating the most recent draft of CityGML 3.0 (Kutzner et al., 2020), ISO TC211 geographic information standards[3], and a semantic web spatio-temporal query framework.

---

[1] https://www.ogc.org/
[2] https://www.ogc.org/standards/citygml

[3] https://www.isotc211.org/

2. A data transformation approach for the generation of versioned CityGML 3.0 datasets as interoperable semantic RDF graphs.

3. Implementation of these graphs in a spatio-temporal triple store to facilitate geospatial and temporal queries on real-world city data.

The remainder of this paper is structured as follows: Section 2 will describe related works in multidimensional semantic city models and spatio-temporal urban graph implementations; Section 3 will detail the proposed integration approach; Section 4 is dedicated to the implementation of the approach on a real-world city dataset and to discuss its strengths and limitations; Section 5 will conclude the proposed contribution, discuss future works on this research, and provide resources for reproducing the work.

2. RELATED WORKS

Although research in geometric and semantic data integration of CityGML with Building Information Model (BIM) standards like Industry Foundation Classes (IFC)[4] has had much evolution (Deng et al., 2016), temporal 4D semantic city model integration remains a large area of research. As previously stated, many existing standards lack rich, native temporal support, and generally require one of two types of integration: either a data model is extended to integrate temporal data formats or multiple data models are integrated as a single, spatio-temporal data model. This data integration can also take place at two levels: data model integration at the conceptual, physical, or implementation level and integration at the model instance or dataset level. Taking this into consideration, some existing works propose approaches to facilitate this integration for enriching 3D semantic city models.

2.1 4D Semantic Urban Data Model Extension

Extending a 3D data model requires starting with a strong, semantic data model. Since its adoption by the OGC in 2012, CityGML has become a commonly used data model standard for this purpose as it is geospatial, easily extensible through Application Domain Extensions (ADEs), and provides a modular semantic conceptual model for various domains of the urban landscape. In the case of CityGML based 4D city models, the conceptual model is traditionally extended to allow better temporal data support (Chaturvedi et al., 2017). Some recent works allow the representation of concurrent and successive versions of city models with graph-based versioning frameworks (Samuel et al., 2020) while others allow for time series and time-dependent data representations (such as internet of things sensor data) (Biljecki et al., 2018).

These extensions are usually created manually and may be time-consuming to generate, verify, and update. Due to the utility demonstrated by these proposed models, the next version of CityGML 3.0 standard will incorporate parts of these works as new Versioning and Dynamizer modules.

2.2 Ontology-Based 4D Urban Data Integration

When creating a spatio-temporal data model from two or more sources, often a common data format is used to facilitate the

integration. For this purpose, much research and official standards have been made for implementing 3D and 4D geospatial data using semantic web technologies, such as RDF (Resource Description Framework) and OWL (Web Ontology Language) (Bonduel et al., 2019, Brink et al., 2014, Lemmens et al., 2016, Métral and Falquet, 2018, Nuninger et al., 2020, Psyllidis, 2015). These proposals use machine-readable description logics to create ontologies for semantic modeling and integration.

Furthermore, network of ontology-based integration approaches for 4D semantic data integration have also been proposed (Hor et al., 2018, Tran et al., 2020, Psyllidis, 2015). These approaches are modular as each ontology integrated into the network can describe a single domain of information or data model. Additionally, extensions to RDF's query language, SPARQL, such as GeoSPARQL[5] and stSPARQL[6] can be integrated to support queries of 2D geometries and 2D geometric and topological relations, with stSPARQL also supporting temporal queries on graph data.

Works have also been proposed for the automatic generation of geospatial and spatio-temporal semantic ontologies from urban data. Some of these approaches use mapping transformations of conceptual urban data models as UML (Unified Modeling Language) to OWL (Brink et al., 2014, De Paepe et al., 2017). These approaches implement their own mappings or propose using the standard ISO 19150-2[7] transformation rules for geospatial data. Other approaches utilize mappings from physical urban data models as XML Schema to OWL (Vinasco-Alvarez et al., 2020, Vinasco-Alvarez et al., 2021, Usmani et al., 2020). These approaches all incur some amount of semantic loss during transformation. In general, UML to OWL transformations may have more well-defined mappings as both modeling languages use similar concepts like classes and properties. However, even the ISO 19150-2 standard can be ambiguous (OGC, 2017) and XML Schema may provide better definitions of spatial semantics (Brink et al., 2014).

2.3 Summary

In summary, most of these proposals demonstrate effective approaches towards integrating 4D semantic city data for 4D semantic city model enrichment. Implementing strong data standards in these models and preserving their interoperability is key during integration, which requires minimizing semantic loss during transformation. Currently, the upcoming version of CityGML 3.0 is promising to be one of the most popular data models in this context. This paper proposes a model-centric integration approach structured around CityGML 3.0, in its latest draft, and other spatio-temporal data standards. This approach will incorporate network of ontologies integration and automated ontology generation approaches to limit semantic data loss of the original conceptual data models. It will also demonstrate how this model can be used to generate interoperable 4D semantic graphs.

3. MODEL-CENTRIC DATA INTEGRATION FOR LIMITING SEMANTIC DATA LOSS

This section will define key concepts and the problem, introduce the proposed approach, and demonstrate its implementation.

---

[4] https://technical.buildingsmart.org/standards/ifc/

[5] https://www.ogc.org/standards/geosparql/

[6] http://www.strabon.di.uoa.gr

[7] https://www.iso.org/standard/57466.html

3.1 Problem Definition

Within 4D semantic city models, the entities concerned are often referred to as city objects. These objects can represent buildings, roads, landmarks, and many other urban entities, and are composed of 3 types of data:

- *Geospatial*: the object's 2D and/or 3D geometric representation, combined with its terrestrial location, often defined by coordinate reference systems.
- *Temporal*: the period when the object exists, represented by two points in time referencing the objects creation and destruction.
- *Semantic*: data that can describe, categorize, or provide context to the object and its components or their relationships to other objects.

Integration of geospatial and temporal data are the most straightforward as the values that represent these data are real numbers and timestamps. Their integration may require conversion but is only prone to a loss of numeric precision (e.g. loss of geometric or geospatial precision during the conversion of coordinate reference systems (Seeger, 2005)). On the contrary, integrating semantic urban data – either as data models or data – can be more challenging.

Semantic integration at the conceptual level can be done by combining or mapping concepts from 2 or more conceptual models. For this, each conceptual model must be represented by the same modeling language to facilitate integration. Since different modeling languages have varying amounts of expressivity it is possible that a concept may not be able to be represented in its entirety or must be represented differently after transformation. This implies a loss of interoperability and data based on the new model may also share this loss. In this paper, these results are referred to as semantic loss. Once the conceptual models are represented in a homogeneous modeling language, equivalent concepts can be related to one another to integrate the models. Once this conceptual integration is complete, integrated 4D semantic datasets can be generated based on this model to enrich 4D semantic city models.

The following section will introduce the proposed integration methodology for limiting semantic loss in 4D semantic urban data integration through 4D semantic graphs.

3.2 4D Semantic Model Integration

For the integration of different 4D semantic urban conceptual models, the approach uses a network of ontologies as a semantic model. To this end the integrated model will be constructed from several existing conceptual models:

- CityGML 3.0 UML model's most recent draft, in particular the Building and Versioning modules for their urban and spatio-temporal semantics.
- GML 3.2 model represented through the ISO 19107 standard and GeoSPARQL's GML 3.2 ontology to support CityGML's geometric and geospatial semantics.
- GeoSPARQL and stRDF ontologies to provide a spatio-temporal query framework for semantic graph interaction.

These models were chosen for their synergy as a network of ontology integration approach. Most of these models have
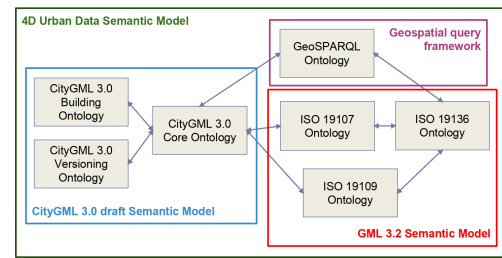

Figure 1. 4D Urban Data Semantic model as a network of ontologies

existing ontological representation in OWL 2 except for the CityGML conceptual model. However, as CityGML is based on GML 3.2 its integration into ontology network is straightforward. The global structure of the network of ontologies is shown in figure 1. To create this semantic model, first, the current CityGML 3.0 conceptual model is converted to OWL 2, then each ontology is aligned to construct the network, and finally, OWL description logic reasoners such as Pellet 2.2.0[8] and HermiT 1.4.3[9] are used to verify the ontology network as logically consistent.

3.2.1 CityGML 3.0 draft transformation to OWL To generate a CityGML 3.0 ontology either the conceptual UML model or the more physical XML Schema are potential candidates for mapping to OWL. There are arguments for choosing each data model and either choice can generate a vastly different ontological structure. The benefit of mapping the XML Schema is that it thoroughly describes the semantics of CityGML documents. But since the goal rests in the preservation of semantics and interoperability of the semantic model, the UML data model is chosen as UML more closely resembles the OWL modeling language. Additionally, automated transformation tools based on the official UML to OWL mappings detailed in ISO 19150-2:2015 such as ShapeChange[10] can be implemented to create OWL ontologies from geospatial UML models. However, implementing this tool for this purpose has been tested by the OGC according to the Testbed 12 ShapeChange Engineering Report (OGC, 2017) and several ambiguities in the ISO 19150-2 standard have been identified which affect the CityGML 3.0 UML model to OWL transformation. The most notable of these being the ambiguous mapping of unions to OWL, which affects CityGML elements such as the union core:cityModelMember. This union groups several important properties of CityGML 3.0 city models, that reference city objects, versions, and version transitions of the model. To overcome this ambiguity, the Testbed 12 report proposes 3 ShapeChange configurations, of which the "flattening" approach is used for transforming unions:

For each UML property or attribute A – that has a union as its value type – a new property is created for each property option B of the union. The name of the new property is constructed by merging the names of A and B with a union separator as follows:

$$\text{"}[Property\ A][Separator][Property\ B]\text{"} \qquad (1)$$

The results of this mapping are shown in figure 2. While this ap-

---

[8] https://www.w3.org/2001/sw/wiki/Pellet
[9] http://www.hermit-reasoner.com/
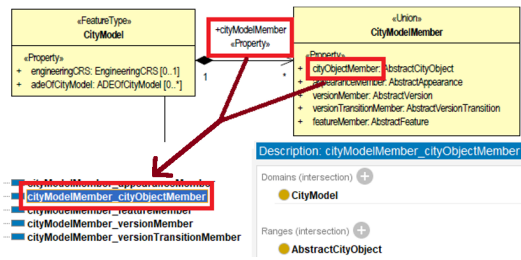[10] https://shapechange.net/

Figure 2. UML Union Flattening example with core:cityModelMember and core:cityObjectMember (top) and their mappings in OWL (bottom)

proach has the undesirable effect of losing the semantic concept of a union, its representation in OWL more closely resembles CityGML 3.0 physical data model – since only the properties of the union are instantiated in CityGML datasets – and thus provides a better semantic model for generating interoperable semantic graphs.

It is also important that the relationships between UML classes and their properties are preserved in the generated ontology. This is ensured by configuring ShapeChange with three mapping transformation rules:

1. Universal quantifications are used in the definition of classes to refer to any properties they may have.
2. Local naming conventions are used to distinguish between different properties of the same name.
3. The domain and range of properties are always declared.

This results in bidirectional references between classes and properties which is useful for automatic semantic graph generation and helps avoid inconsistencies when validating the ontology.

Finally, ShapeChange is configured to map any geometric references to their equivalent in the GeoSPARQL and ISO standard ontologies. Any other configurations follow the OGC Testbed 12 recommendations, enabling ISO 19150-2 rules whenever possible. The final output of the transformation results in 17 linked ontologies, one for each module of the CityGML 3.0 Conceptual Model. Figure 3 exemplifies a mapping between the UML representation of a version and its representation in OWL 2. Next, links are made manually between the ISO 19107 and GeoSPARQL ontologies. Once validated, this is considered as a 4D semantic urban data network of ontologies.

3.3 Graph Generation of 4D Semantic City Models

In this section, the second part of the proposed methodology is demonstrated: how to generate 4D semantic datasets from the integrated network of ontologies. To do this, the class and properties assertions (or the TBox) of the ontology network are queried to generate 4D urban data graphs as instances of these assertions (or the ABox). First, these queries must answer the following questions of an atomic datum within a 4D urban dataset:

- Which Class, Property, or Datatype in the network defines the datum?
- Does the datum represent geometry?
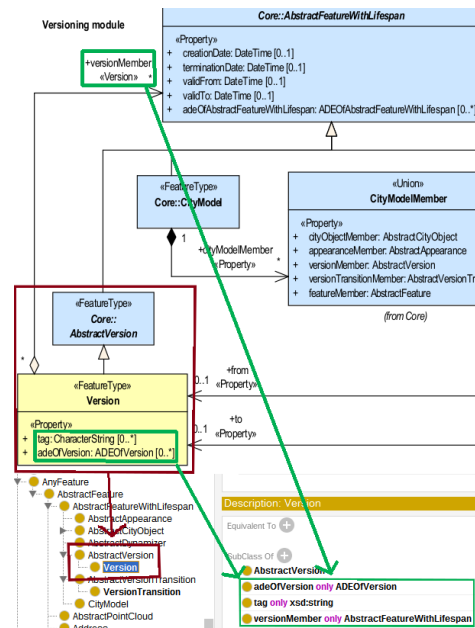- Does the datum have temporal properties?

Figure 3. CityGML 3.0 vers:Version UML (top) to OWL Class (bottom)

SPARQL queries can be implemented to answer these questions. For simplification, a city model as a CityGML 3.0 XML document is considered to be a set of nodes, where each node may have parents and/or children. Determining if a node can be a class or property instance is done using the node's XML tag and searching if a class or property assertion of the same identifier is in the network. Query 1 shows how this can be done for verifying if a node is an object property. This query uses a node's identifier "$ntype$" and its parent's identifier, "$ptype$", to find properties with "$ptype$" in their domain (line 5) or universal quantifications with "$ntype$" asserted by class restrictions (line 8). A regex function is used to filter these properties with local naming conventions (line 15). Once these types of queries are defined, they can be used to map city model datasets to the ABox of the ontology network as shown in algorithm 1.

Query 1 Find a ObjectProperty Assertion with local naming conventions

---

```
 1: SELECT DISTINCT ?property
 2: WHERE {
 3:    { ?property a owl:ObjectProperty ;
 4:         rdfs:domain ?domain .
 5:      ptype rdfs:subClassOf* ?domain .
 6:    }
 7:    UNION
 8:    { ptype a owl:Class ;
 9:         rdfs:subClassOf [
10:          a owl:Restriction ;
11:          owl:allValuesFrom ?someClass ;
12:          owl:onProperty ?property
13:        ] .
14:    }
15:    FILTER regex(STR(?property), "ptype. * .ntype$")
16: }
```

Algorithm 1 looks through all nodes of a given city model in Line 1. For all nodes which are defined by class assertions (line 2) a triple will be created according to if the children of the node (line 4) are defined by a class assertion (line 5), an objectProperty assertion (line 8), a datatypeProperty assertions (line 14), or a datatype assertion (line 21). All of these cases will add the

triple (lines 11, 16, 19, 23) to a graph to be returned in line 28.

Algorithm 1 Semantic Graph Generation

Input: A set of ontological entities, $O = \{o_1, o_2, ...o_n\}$, a set
of tree nodes, $N = \{n_1, n_2, ...n_m\}$
Output: A set of triples $\{\langle s_1, p_1, v_1 \rangle, \langle s_2, p_2, v_2 \rangle, ...\langle s_n, p_n, v_n \rangle\}$
1: **let** $graph$ be an empty set, $\{\}$
2: **for** n in N **do**
3:    **if** O.hasClass(n) **then**
4:       **for** c in n.children **do**
5:          **if** O.hasClass(c) **then**
6:             $property = O$.findPropertyBetween(n, c)
7:             $graph$.push($\langle n.name, property.name, c.name \rangle$)
8:          **else if** O.hasObjectProperty(c) **then**
9:             **for** gc in c.children **do**
10:                **if** O.hasClass(gc) **then**
11:                   $graph$.push($\langle n.name, c.name, gc.name \rangle$)
12:                **end if**
13:             **end for**
14:          **else if** O.hasDatatypeProperty(c) **then**
15:             **if** c.text **then**
16:                $graph$.push($\langle n.name, c.name, c.text \rangle$)
17:             **end if**
18:             **for** gc in c.children **do**
19:                graph.push($\langle n.name, c.name, gc.name \rangle$)
20:             **end for**
21:          **else if** O.hasDatatype(c) **then**
22:             $property = O$.findPropertyBetween(n, c)
23:             $graph$.push($\langle n.name, property.name, c.text \rangle$)
24:          **end if**
25:       **end for**
26:    **end if**
27: **end for**
28: **return** $graph$

During graph generation of CityGML datasets, datatypes from the GeoSPARQL standard are used to represent geometric and geospatial data. To determine when an XML node represents geometry in GML 3.2, GeoSPARQL and stRDF's rules for declaring geo:gmlLiteral values can be implemented according to the definition:

Valid geo:gmlLiterals are formed by encoding geometry information as a valid element from the GML schema ... in GML 3.2.1 this is every element directly or indirectly indirectly in the substitution group of the element {http://www.opengis.net/ont/gml/3.2}Abstract-Geometry. (OGC, 2012)

By this definition if an XML node is an instance of a subclass of geo:AbstractGeometry in the ontology network, the XML node represents geometry and its 2D geometric representation can be stored in a triple in the graph. The value of the triple being a copy of the XML data of the original node and its descendant nodes encoded as a literal string. To represent temporal data, xsd:dateTime values are used to represent a point in time. In CityGML 3.0 every city object can use its core:creationDate and core:terminationDate properties as temporal points to representing the beginning and end of its existence.

4. IMPLEMENTATION AND RESULTING DATASETS

4.1 Methodology Implementation

Under the Urban Data Services and Visualization Project (UD-SV)[11], a transformation tool, UD-Graph[12], was implemented to

---

[11] https://github.com/VCityTeam/UD-SV
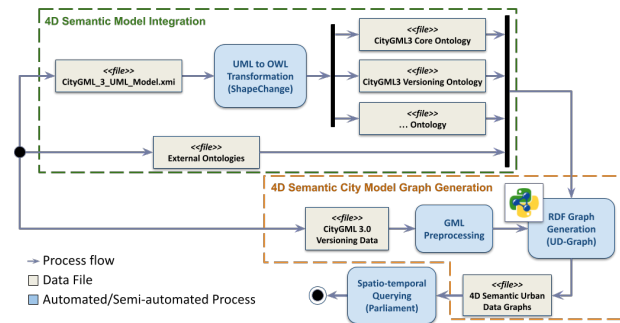[12] https://github.com/VCityTeam/UD-Graph

---

Figure 4. Proposed 4D Semantic Model Integration and Graph Generation Pipeline

apply this methodology to generate 4D semantic graphs as RDF from CityGML 3.0 data using the RDFLib[13] Python library. As shown in figure 4, the methodology proposed in section 3.2 and 3.3 is illustrated. An additional preprocessing step is necessary before graph generation to prepare GML data for use with Geo-SPARQL's 2D queries. This step can add coordinate-reference-system declarations if necessary and flattens 3D GML geometry to 2D. During generation, gml:id attributes are used as triple identifiers whenever possible. This also applies to Xlinks which are used to link CityGML 3.0's versioning data – such as a city model's version to the corresponding city objects of that version. Afterward, these 4D semantic urban graphs were loaded into a Parliament[14] triple store to implement spatio-temporal queries through GeoSPARQL.

The following section will detail how UD-Graph and this methodology are applied to an initial 4D semantic urban dataset.

4.2 Resulting Dataset

The 4D semantic urban datasets that are used in these experiments originate from the 1st district of the city of Lyon, France[15] as CityGML 2.0. The building and geospatial data from these datasets are converted to CityGML 3.0 with a CityGML 2 to 3 open-source conversion tool[16] and then enriched with temporal versioning data and stored on the CityGML 3.0 Encoding Github[17]. Using the proposed transformation methodology, an initial 4D semantic city model dataset was created. Figure 5 illustrates a historical succession of 2 versions of a city model containing 4 buildings. A version transition between these two versions is composed of 3 transactions (or changes): a building deletion ($Transaction\_1$), a building replacement ($Transaction\_2$), and a building insertion ($Transaction\_3$). Figure 6 shows urban data classes and class instances generated from the dataset according to the CityGML building and core ontologies (highlighted blue in figure 1) and the GML geospatial ontologies (highlighted red in figure 1), visualized by OntoGraph[18].

Query 2 returns all the buildings that have a physical existence intersecting a given point of time. Lines 5-7,9-11 define the creation and termination dates of a building, i.e., the time period of its physical existence. Lines 8 and 12 filter the buildings that

---

[13] https://rdflib.readthedocs.io/
[14] https://github.com/SemWebCentral/parliament
[15] https://data.grandlyon.com/
[16] https://github.com/tum-gis/citygml2-to-citygml3
[17] https://github.com/opengeospatial/CityGML-3.0Encodings/tree/master/CityGML/Examples
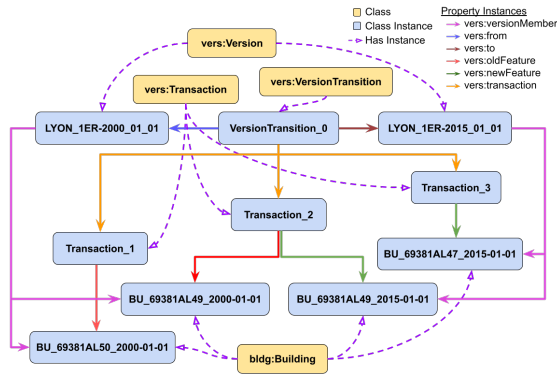[18] https://protegewiki.stanford.edu/wiki/OntoGraf

---

Figure 5. Diagram of an RDF Graph of a Historical Succession Transition of Between 2 City Model Versions (above) with several Buildings (below)

exist during the temporal point specified by the user (lines 3,4). In this example, the query looks for buildings that existed on 01/01/2010. Also, note that the properties in the query use local naming conventions as mentioned in section 3.2.1.

Query 2 Find all buildings that existed at a temporal point

```
 1: SELECT DISTINCT ?building ?tp1 ?tp2
 2: WHERE {
 3:    BIND("2010-01-01T00:00:00"^^xsd:datetime
 4:       as ?point)
 5:    ?building a bldg:Building ;
 6:       core:AbstractFeatureWithLifespan.creationDate
 7:          ?tp1 .
 8:    FILTER(?tp1 ≤ ?point)
 9:    OPTIONAL { ?building
10:       core:AbstractFeatureWithLifespan.terminationDate
11:          ?tp2 .
12:       FILTER(?point ≤ ?tp2)
13:    }
14: }
```

Query 3 returns all the city objects in a version, *v1*, of a concurrent point of view. Line 3 verifies whether the version specified by the user is indeed a version and Line 4 and 5 return all the city object members and type for this given version.

Query 3 Find all city objects in a city model version

```
 1: SELECT DISTINCT ?cityobjectmember ?type
 2: WHERE {
 3:    v1 a vers:Version ;
 4:       vers:Version.versionMember
 5:          ?cityobjectmember .
 6:    ?cityobjectmember a ?type .
 7: }
```

4.3 Discussion

This initial approach integrates standardized conceptual models of spatio-temporal and urban data to create integrated 4D semantic datasets as a first step toward 4D semantic city model enrichment. Unlike approaches based on extending the CityGML 2.0 model to support spatio-temporal data, the CityGML 3.0 conceptual model is rich enough to represent 4D semantic city models without extension. By implementing automated conversion tools such as ShapeChange, ISO standard generation of CityGML 3.0 as a ontology network is possible and the benefits of ontology based integration approaches can be taken advantage of with minimal semantic data loss.

The CityGML 3.0 model is especially synergistic with this approach, regarding the creation of 4D semantic data, since the OGC generates the physical data model as XML Schema directly from the UML model using ShapeChange. Therefore, class and properties names from the UML model closely resemble their XML representation and thus transformation mappings from CityGML 3.0 to semantic urban graphs lose minimal semantic data and rest interoperable with their original conceptual model.

In addition, semantic graph formats like RDF align well with CityGML 3.0's graph-like representations of city models in the Versioning module. By integrating geospatial query frameworks like GeoSPARQL spatial navigation of these 4D semantic urban data graphs is possible. This can have many potential applications like change detection of city objects between versions of a city model which may rely on graph formats (Nguyen and Kolbe, 2020) and smart city applications based on Semantic Web technologies (Gaur et al., 2015, Bischof et al., 2014).

However, there are two modeling limitations of this approach to be addressed in future works. First and foremost, while the CityGML 3.0 application schema is directly generated from the conceptual model, the GML 3.2 model does share this characteristic. Many discrepancies in naming conventions between the GeoSPARQL ontology, ISO 19107 models, and GML 3.2 data were identified. For example, the application schema uses gml:id as a unique local identifier for geometric entities, while this does not exist in the GeoSPARQL ontology and ISO 19107 uses featureID as its identifier. To solve this, additional mappings are being integrated into the ontology network based on ISO TC211 standards to GML 3.2 instance mappings implemented by Enterprise Architect[19], an ISO standard compliant UML modeling platform.

Secondly, as the generated CityGML 3.0 ontologies are constructed from UML's frame-based, "closed-world" assumption of the conceptual model, the mapping transformations to OWL's more "open-world" assumption of the conceptual model are subjective and require some interpretation (Brink et al., 2014, Cox, 2013). These ontologies are defined according to a more restrictive interpretation in order to guard as many semantic relationships as possible. Because of this, it is possible that while this ontology network works well to generate 4D semantic graphs, other applications of UML to OWL mappings may require a more "open-world" interpretation depending on the purpose of the ontologies.

In addition, to ameliorate the temporal queries proposed in this paper, spatio-temporal query frameworks such as stSPARQL can be implemented to implement more complex temporal queries and improve performance (Garbis et al., 2013).

5. CONCLUSIONS

The evolving urban analysis landscape requires the integration of multiple heterogeneous and autonomous models catering to describing the urban lifecycle. However, a direct translation of data in different evolving data formats to the desired format may lead to semantic data loss. This article presents a methodology for integrating 4D semantic urban data models as a network

---

[19] https://sparxsystems.com/enterprise_architect_user_guide/15.2/ model_domains/uml_profile_for_gml.html
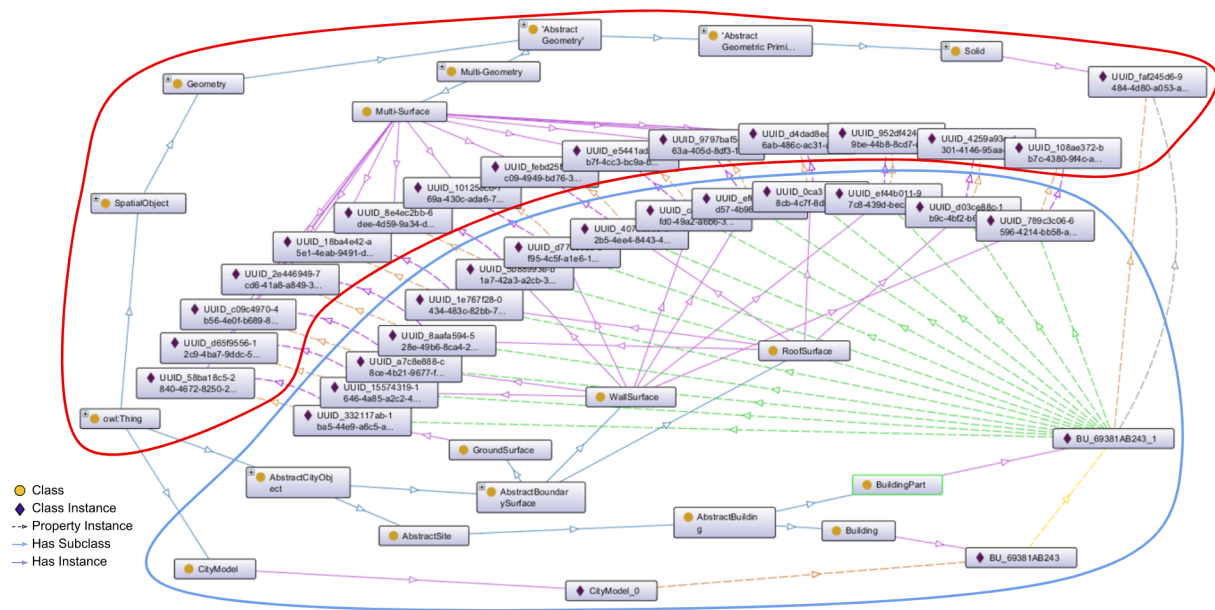
Figure 6. RDF Graph of classes and instances concerning building data generated according to the 4D Urban Data Semantic model defined in Figure 1; CityGML 3.0 Building data is circled in blue; GML geospatial data is circled in red; dashed arrows represent instances of semantic properties (or relationships)

of ontologies. This UML-based transformation approach based on standards helps to preserve interoperability and reduce semantic loss. This paper also proposes a pipeline for generating integrated RDF data in conformance to this model. The data obtained with this approach were validated and spatio-temporal queries were tested and executed.

Future courses of action include improving the existing semantic model and integrating more urban data models like cadastral data, city documents, and concurrent points of view of urban evolution. Also, the addition of bidirectional transformations could enable RDF to CityGML transformations for non-RDF ready applications. And finally, the implementation of more complex 3D and 4D urban data queries with SPARQL extensions like those provided by stSPARQL and BimSPARQL (Zhang et al., 2018) are being explored.

6. DATA AND REPRODUCIBILITY

Detailed notes and the code for reproducing the results of section 3 and 4 can be found on Software Heritage here[20]. The generated CityGML 3.0 Versioning dataset is located under the folder */Transformations/test-data/RDF/OWL-based-transformations*.

---

[20] https://archive.softwareheritage.org/swh:1:dir:
e3df50dc7b066ce284ca66c1633e4eb5c2aadf57;origin=
https://github.com/VCityTeam/UD-Graph;visit=swh:1:snp:
c6ac42c1540a27fbfcdaf09db305d6949751700d;anchor=swh:1:
rev:2b02e0c494072dbe4eb6f4d3567c661ae911b2a7

[21] https://projet.liris.cnrs.fr/vcity/

REFERENCES

Batty, M., 2018. Digital twins. *Environment and Planning B: Urban Analytics and City Science*, 45(5), 817–820.

Biljecki, F., Kumar, K., Nagel, C., 2018. CityGML Application Domain Extension (ADE): overview of developments. *Open Geospatial Data, Software and Standards*, 3(1), 13.

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(4), 2842–2889.

Bischof, S., Karapantelakis, A., Nechifor, C.-S., Sheth, A. P., Mileo, A., Barnaghi, P., 2014. Semantic Modelling of Smart City Data. W3C, Berlin, Germany.

Bonduel, M., Wagner, A., Pauwels, P., Vergauwen, M., Klein, R., 2019. Including widespread geometry formats in semantic graphs using RDF literals. *Proceedings of the 2019 European Conference for Computing in Construction*, European Council on Computing in Construction, 341–350.

Brink, L., Janssen, P., Quak, W., Stoter, J., 2014. Linking spatial data: automated conversion of geo-information models and GML data to RDF. *International Journal of Spatial Data Infrastructures Research*, 9, 59–85.

Chaturvedi, K., Kolbe, T., 2019. A Requirement Analysis on Extending Semantic 3D City Models for Supporting Time-Dependent Properties. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W9, 19–26.

Chaturvedi, K., Smyth, C. S., Gesquière, G., Kutzner, T., Kolbe, T. H., 2017. Managing Versions and History Within Semantic 3D City Models for the Next Generation of CityGML. A. Abdul-Rahman (ed.), *Lecture Notes in Geoinformation and Cartography*, Advances in 3D Geoinformation, Springer, 191 – 206.

Claramunt, C., 2020. Ontologies for geospatial information: Progress and challenges ahead. *Journal of Spatial Information Science*, 35–41.

Cox, S. J. D., 2013. An explicit owl representation of iso/ogc observations and measurements. *Proceedings of the 6th International Conference on Semantic Sensor Networks - Volume 1063*, SSN'13, CEUR-WS.org, Aachen, DEU, 1–18.

De Paepe, D., Thijs, G., Buyle, R., Verborgh, R., Mannens, E., 2017. Automated uml-based ontology generation in oslo2. E. Blomqvist, K. Hose, H. Paulheim, A. Ławrynowicz, F. Ciravegna, O. Hartig (eds), *The Semantic Web: ESWC 2017 Satellite Events*, Springer International Publishing, Cham, 93–97.

Deng, Y., Cheng, J. C., Anumba, C., 2016. Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison. *Automation in Construction*, 67, 1–21.

Euzenat, J., 2014. The category of networks of ontologies. Research Report RR-8652, INRIA. hal-01093207.

Garbis, G., Kyzirakos, K., Koubarakis, M., 2013. Geographica: A benchmark for geospatial rdf stores (long version). H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. Noy, C. Welty, K. Janowicz (eds), *The Semantic Web – ISWC 2013*, Springer Berlin Heidelberg, Berlin, Heidelberg, 343–359.

Gaur, A., Scotney, B., Parr, G., McClean, S., 2015. Smart City Architecture and its Applications Based on IoT. *Procedia Computer Science*, 52, 1089–1094.

Gröger, G., Plümer, L., 2012. CityGML – Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71, 12–33.

Hor, A.-H., Sohn, G., Claudio, P., Jadidi, M., Afnan, A., 2018. A Semantic Graph Database for BIM-GIS Integrated Information Model for an Intelligent Urban Mobility Web Application. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4, 89–96.

Jaillot, V., Servigne, S., Gesquière, G., 2020. Delivering time-evolving 3D city models for web visualization. *International Journal of Geographical Information Science*, 34(10), 2030–2052.

Julin, A., Jaalama, K., Virtanen, J.-P., Pouke, M., Ylipulli, J., Vaaja, M., Hyyppä, J., Hyyppä, H., 2018. Characterizing 3D City Modeling Projects: Towards a Harmonized Interoperable System. *ISPRS International Journal of Geo-Information*, 7(2), 55.

Kolbe, T., 2009. Representing and Exchanging 3D City Models with CityGML. *3D Geo-Information Sciences*, 15–31.

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.

Lemmens, R. L. G., Falquet, G., Métral, C., 2016. Towards Linked Data and ontology development for the semantic enrichment of volunteered geo-information. *Proceedings of: Link-VGI: LINKing and analyzing Volunteered Geographic Information (VGI) across different platforms, workshop at the AGILE 2016 Conference*.

Métral, C., Falquet, G., 2018. Extension and Contextualization for Linked Semantic 3D Geodata. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4-W10, Copernicus GmbH, 113–118.

Nguyen, S. H., Kolbe, T. H., 2020. A Multi-Perspective Approach to Interpreting Spatio-Semantic Changes of Large 3D City Models in CityGML Using a Graph Database. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, VI-4-W1-2020, Copernicus GmbH, 143–150.

Nuninger, L., Verhagen, P., Libourel, T., Opitz, R., Rodier, X., Laplaige, C., Fruchart, C., Leturcq, S., Levoguer, N., 2020. Linking Theories, Past Practices, and Archaeological Remains of Movement through Ontological Reasoning. *Information*, 11(6), 338.

OGC, 2012. GeoSPARQL - A Geographic Query Language for RDF Data | OGC.

OGC, 2017. OGC 16-020: Testbed-12 ShapeChange Engineering Report.

Psyllidis, A., 2015. Ontology-Based Data Integration from Heterogeneous Urban Systems: A Knowledge Representation Framework for Smart Cities. *CUPUM 2014: Proceedings of the 14th International Conference on Computers in Urban Planning and Urban Management*, MIT, Cambridge, USA.

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.

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(1), 99–112.

Seeger, H., 2005. Spatial referencing and coordinate systems. *New Developments in Geographical Information Systems: Principles, Techniques, Management and Applications*, 1, 427–436.

Tran, B.-H., Aussenac-Gilles, N., Comparot, C., Trojahn, C., 2020. Semantic Integration of Raster Data for Earth Observation: An RDF Dataset of Territorial Unit Versions with their Land Cover. *ISPRS International Journal of Geo-Information*, 9(9), 503.

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.

Vinasco-Alvarez, D., Samuel, J. S., Servigne, S., Gesquière, G., 2020. From CityGML to OWL. Technical Report, LIRIS UMR 5205.

Vinasco-Alvarez, D., Samuel, J. S., Servigne, S., Gesquière, G., 2021. Towards a semantic web representation from a 3d geospatial urban data model. *SAGEO 2021, 16th Spatial Analysis and Geomatics Conference*, 227–238.

Zhang, C., Beetz, J., Vries, d., 2018. BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data. *Semantic Web*, 1–27.