Smart Campus based on Linked Data

PhD thesis

Barnabás Szász

Supervisor: Dr. András Micsik

UNIVERSITY OF DEBRECEN
Doctoral School of Informatics

Debrecen, 2019

Okos egyetemi szolgáltatásokat megalapozó kapcsolt adathalmazok

Egyetemi doktori (PhD) értekezés

Szász Barnabás

Témavezető: Dr. Micsik András

DEBRECENI EGYETEM
Természettudományi és Informatikai Doktori Tanács
Informatikai Tudományok Doktori Iskola

Debrecen, 2019

Ezen értekezést a Debreceni Egyetem Természettudományi Doktori Tanács a **Informatikai Tudományok Doktori Iskola Elméleti számítástudomány, adatvédelem és kriptográfia doktori** programja keretében készítettem a Debreceni Egyetem természettudományi doktori (PhD) fokozatának elnyerése céljából.

Nyilatkozom arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.

Debrecen, 2019.

…………………………

a jelölt aláírása

Tanúsítom, hogy **Szász Barnabás** doktorjelölt **2014-2019** között a fent megnevezett Doktori Iskola **Elméleti számítástudomány, adatvédelem és kriptográfia doktori** programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Nyilatkozom továbbá arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét. Az értekezés elfogadását javasolom.

Debrecen, 2019.

…………………………

a témavezető aláírása

OKOS EGYETEMI SZOLGÁLTATÁSOKAT MEGALAPOZÓ KAPCSOLT ADATHALMAZOK
SMART CAMPUS BASED ON LINKED DATA

Értekezés a doktori (Ph.D.) fokozat megszerzése érdekében

Írta: **Szász Barnabás** okleveles Programtervező Matematikus

Készült a Debreceni Egyetem **Informatikai Tudományok Doktori Iskolája** (**Elméleti számítástudomány, adatvédelem és kriptográfia Doktori program** programja) keretében

Témavezetők:
Dr. Micsik András
Dr. Bognár Katalin

A doktori szigorlati bizottság:
elnök: ………………………… …………………………
tagok: ………………………… …………………………
………………………… …………………………
A doktori szigorlat időpontja: …………………………
Az értekezés bírálói:
………………………… …………………………
………………………… …………………………

A bírálóbizottság:
elnök: ………………………… …………………………
tagok: ………………………… …………………………
………………………… …………………………
………………………… …………………………
………………………… …………………………

Az értekezés védésének időpontja: ……………… … .

Table of Content

1 Introduction 1

  1.1 Research objective and contributions 2

  1.2 Thesis structure 7

2 Basic concepts and definitions 8

  2.1 Semantic Web 8

  2.2 SPARQL 9

  2.3 SPARQL UPDATE protocol 11

  2.4 Ontologies 12

  2.5 Linked Data 15

3 Linked Data Enrichment with Self-Unfolding URIs 18

  3.1 Linked Data Enrichment 19

  3.2 Use cases 21

  3.3 Generic scenarios 24

  3.4 Implementation 26

  3.5 Conclusion 32

4 The OLOUD Ontology 34

  4.1 Introduction 34

  4.2 Linked Data consumption use cases 37

  4.3 Ontologies for education 39

  4.4 Methodology 47

  4.5 Integration with other ontologies 49

4.6 Ontology description 50
4.7 Example data 53
4.8 Conclusions 57
5 The iLOC Ontology 58
5.1 Modeling Indoor Spaces with Ontologies 58
5.2 Ontology specification 60
5.3 Integration with other ontologies 64
5.4 Extension points 65
5.5 Conclusion 66
6 Indoor Navigation with distributed Linked Data 67
6.1 Graph route-finding tools 67
6.2 The iLOC ontology in use 70
6.3 Navigation 71
6.4 Finding and navigating to a specific nearby POI 75
6.5 Query Performance 76
6.6 Conclusion 77
7 Indoor modeling best practices with iLOC 78
7.1 Indoor modeling approaches 78
7.2 Indoor modeling with iLOC 78
7.3 The Óbuda University use case 84
7.4 Patient navigation in hospitals 86
7.5 The Budapest Airport Terminal use case 88
7.6 Conclusion 89
8 Summary 90
8.1 Summary of the work 90
8.2 Future work 93
8.3 Conclusion 93
9 Összefoglalás 94

9.1 Kutatási eredmények 94
Appendix A. 101
Appendix B. 106
SPARQL examples 106
Appendix C. 110
Ontology for Indoor Navigation 110
OLOUD: Ontology for Linked Open University Data 115
Appendix D. 120
A List of papers of the author and citations to them 120
References 123

Figures

Figure 1: The Semantic Web Layer Cake 8

Figure 2: The size of the Linked Open Data cloud in 2018 17

Figure 3: Major concepts in the Hungarian higher education 35

Figure 4: Overview of the main classes and properties in OLOUD 50

Figure 5: Extract of the dataset in the Lodmilla browser 55

Figure 6: The overview of the iLOC ontology 61

Figure 7: The topological model 70

Figure 8: Modeling building structure in iLOC 72

Figure 9: iLOC POI network example 73

Figure 10: Modeling indoor routes with the RouteSection and RouteConstraint classes 74

Figure 11: Hallway entities on the third floor 81

Figure 12: POI-hub in Hallway entity 360a 82

Figure 13: Connecting POI-hubs on the third floor 83

Figure 14: Screenshot of a mobile indoor navigation application developed for the Óbuda University 85

Figure 15: Example routes modelled within a smart hospital 86

Figure 16: A route modeled on the Budapest Airport Terminal 2A 88

Figure 17: Average query times with Fuseki 102

Figure 18: Average query times with Virtuoso 103

Figure 19: Average query time of 3 steps route calculation with different queries 103

Figure 20: Average query times of the 8-step query 104

Tables

Table 1: Description Logic constructs [4] 14

Table 2: Concept coverage of relevant ontologies... 41

Table 3: Core terms mapping between the different education-related ontologies 46

Table 4: Average query times – Fuseki 104

Table 5: Average query times - Virtuoso.. 105

Table 6: Standard deviations in the query times – Fuseki 105

Table 7: Standard deviations in the query times – Virtuoso........................ 105

1 Introduction

Universities around the world are facing nowadays new questions and challenges, while they are working on their strategy. These include: How should universities change to satisfy new demands and needs? What role new technologies play in how students learn, and academics teach? And how are university campuses going to adapt to new technologies whilst maintaining their position as incubators for exploration and innovation? In this new competitive landscape of services, the university campus needs to remain a relevant place of engagement. The focus, therefore, is on making the university a more sustainable place for human exchange and interaction. As in every aspect of modern life, technology is playing a major role in this transition too.

Nowadays, information sharing, and consuming is happening online, using different services, such as email and messaging, file transfer, chat, and video calls and many other, but still, the most popular service is the World Wide Web. The traditional use cases of the WWW were focusing on human users as direct consumers and producers of the content. The data in most Web pages is at best semi-structured [1] and a large percentage of the pages are "hidden" behind form-based interfaces designed for human users [31], resulting in a new set of best practices for publishing and connecting structured data on the WWW. The resulting form of the published data is commonly referred to as Linked Data and has gained tremendous momentum in recent years.

One aspect of the Smart University or Smart Campus concept is to improve the teaching and learning environment using modern data fusion and data consumption techniques. A campus has a large number of people with a substantial set of common information needs [56]. Therefore, it has great importance in this area to establish a common data model which enables the interconnection of fragmented data from heterogeneous data sources.

The overall goal of publishing Linked Data about universities is to open up and connect the rich data sources, both concerning administrative and research data.

1.1 Research objective and contributions

This thesis contributes to the "Smart Campus" topic, researching the opportunities and benefits of publishing university related open linked data in a machine consumable way. In this section, a short overview is provided for each research objective following the main contributions:

1.1.1 Auto-generating URIs for Linked Data resources with composite keys

Linked Data resources are identified by Uniform Resource Identifiers. Although it is widely accepted, that URIs should not carry any semantic meaning, in the practice it provides value, if the URIs are human-readable, so a human being can also easily identify a specific resource in a list of URIs.
It is an important step in any Linked Data project to define the conventions for URI assignments. In some cases, resources already have their natural identifiers to be used as the local part of the resources' URI, or they can be inherited from previous databases. In other cases, resources might not have an inverse-functional property providing a single identifier for their URI, only combination of multiple keys can identify such resources. Typical examples are recurring events without specific titles or other identifiers for each recurrence. These events can be identified by their exact time and location. RDF provides tools to completely omit identifiers and use Blank Nodes, but these have further issues as such resources cannot be referenced directly.

In [67] we elaborated on a mechanism to generate identifiers for Linked Data resources having composite keys and generation of the implicit data represented by such resources using standard SPARQL. The mechanism simplifies the maintenance of triple stores without changing the actual RDF storage.

1.1.2 Holistic university related Linked Data modeling and publishing with ontologies

In the Hungarian practice, university information systems are typically separate and somewhat isolated from each other. There are science management, e-learning, etc. services, but the integration of such services and data sources are still not fully utilized. Currently, there is no completely integrated database in the Hungarian higher education holding and serving all related data. The available data sources are usually acting as closed systems, and although the data itself is public, public access is not granted. Data about courses, students and lecturers are managed in separate systems and the connections between them are not always set explicitly. Study materials for courses and students are not easy to determine. To access implicit relationships, it is worthwhile to model the relevant concepts and their relationships in formal ontologies. Publishing Linked Data requires using a (or multiple) schema to define the meaning of the used terms. Using ontologies to publish Linked Data adds more explicit semantics to the schema and enables inference on the data, thus implicit relationships can be made explicit.

Although there are several ontologies to cover different parts of the university domain, a single ontology would be too complex to cover the whole use case. The presented OLOUD ontology aims to reuse and integrate concepts from several sources to support publishing and consuming personal timetables, navigation and other types of help for students and lecturers. The modeled domains include lectures, semesters and personnel, but also buildings, events and learning related concepts. We summarize issues with the source ontologies, such as missing links, inconsistencies as well as many overlaps between them. OLOUD acts as a glue for a selection of existing ontologies, and thus enables us to formulate SPARQL queries for a wide range of practical questions of university students. The ontology enables the universities to publish data from multiple data sources in a consistent and easy to use manner. [22], [64] and [65] provides further details about our work.

During the work, we realized the need for an ontology supporting indoor navigation and indoor space modeling.

1.1.3 Semantic modeling indoor spaces for publishing as Linked Open Datasets

There are multiple approaches to model indoor spaces. These models are fundamentally different considering their use, cost of modeling and use of the model and the concepts the models are using. Models can be structural, geometric and semantic, amalgamating the properties of structural, geometric and topologic models. Structural descriptions relate rooms and concepts to each other in a hierarchy, not providing the required level of detail for indoor navigation, while geometric descriptions (e.g. floor plans) are too complex, processing and using them for indoor navigation can be troublesome.

Our objective was to provide step-by-step navigation, which required an ontology providing a semantic description, thus the iLOC ontology born. The topologic description is the most accurate supporting step-by-step navigation, which was extended by semantic descriptions (e.g. type of rooms, access level, capability and preference of users, etc.). It can provide routes from point A to B, additionally routes to unknown places described by their semantic properties (e.g. nearest wheelchair accessible restroom or vending machine providing gluten-free sandwiches, etc.).

The objective of our ontology is to support the publishing of indoor location and navigation data as Linked Open Datasets and to support the development of applications using them. We also showcased, that Linked Data can support indoor navigation if proper semantic models are built from the structural and geometric description of the involved buildings using concepts from the iLOC ontology. The ontology is documented in [23], [24], [25] and [66].

1.1.4 Indoor navigation with iLOC graphs based on personal preferences, abilities, and permissions

Outdoor navigation is widely available nowadays and helps people to find a place while driving or walking or using public transport. This type of navigation is usually based on coordinates provided by GPS or Glonass signals. Inside buildings, however, a navigation system must cope with more complex routes using floors, elevators, and staircases in the lack of GPS/Glonass signals. The users might be more limited regarding their abilities, permissions and their preferences could be more fine-grained. The current solutions usually require special and expensive hardware for exact positioning. Thus, opposed to the outdoor scenario, there is no single, accepted basic methodology for navigation. The ontology facilitates locations description and links to relevant data. It can also describe how two places are connected including distance or constraints (e.g. fulfilling wheelchair requirements or needing special badges, etc.). The objective of publishing indoor location data as LOD is to support the development of mobile and web applications using this dataset.

In [23], [24], [25] and [66] we were the first to demonstrate, that using standard SPARQL queries and the iLOC ontology, one can get personalized indoor routes from Linked (Open) Data.

1.1.5 Modeling indoor spaces with the iLOC ontology based on their floorplan

The iLOC ontology provides concepts for the topologic description of indoor spaces, which is the most important part of a semantic indoor description. In a nutshell, with the ontology one can describe a network of connected POIs (Point of Interest).

Semantic indoor description can be derivated from their geometric (e.g. floorplan) and topologic description, omitting certain complexities and dividing up bigger spaces into smaller ones. During the process, the building should be marked with easy to identify POIs and the available direct routes between these. Indoor routes will always lead through multiple POIs marked in such a way.

In [23] we are describing a methodology of dividing buildings into a network of POIs and finding the most optimal routes between these points. In larger rooms (e.g. halls, corridors, theater rooms, etc.), it is important to find the right amount of POIs, allowing navigation to all of the significant portion of such larger spaces. Bigger rooms, stairways, elevators might be accessible from multiple levels within the building, providing access to different floors and increasing the complexity of the model.

In [23] a case study is provided about modeling of the building of the Óbuda University. Here we first model the different building parts and then identify the POIs in and between these building parts. As this process is cumbersome and error-prone, it is important to follow the outlined methodology.

1.2 Thesis structure

The rest of the work is organized as follows:

- Chapter 1 introduces the topic of the dissertation, illustrates the motivations of the theses, and summarizes the main contributions.
- Chapter 2 is devoted to introducing the concepts and terminologies used in the thesis.
- Chapter 3 introduces to the concept of Linked Data Enrichment with Self-Unfolding URIs.
- In Chapter 4, the OLOUD ontology is presented. OLOUD stands for Ontology for Linked Open University Data.
- Chapter 5 introduces the iLOC ontology, which is an ontology supporting Linked Data publication for building modeling and indoor navigation.
- In Chapter 6 further details are given to indoor navigation with Linked Data.
- Chapter 7 elaborates on indoor space modeling best practices with the iLOC ontology.

2 Basic concepts and definitions

This chapter briefly introduces the concepts and terminologies used in this thesis for better understanding.

2.1 *Semantic Web*

The Semantic Web provides a multi-layer method for the formalization of conceptualization in a distributed environment. It is built on established, widely used technologies.

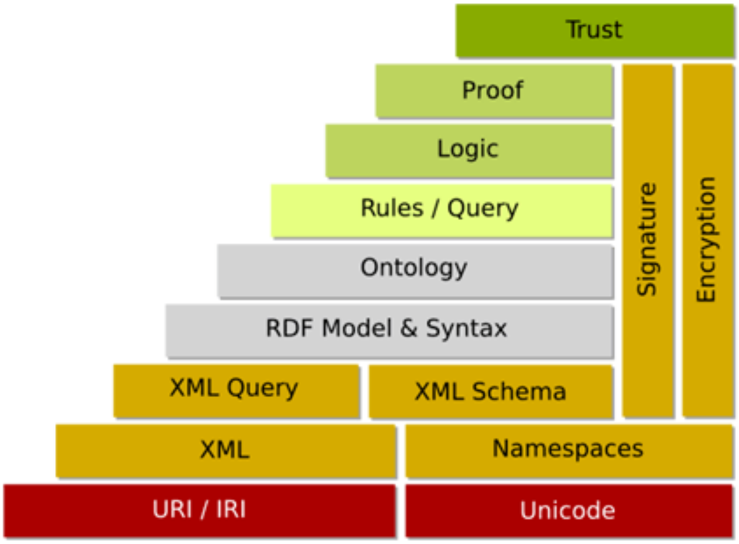


Figure 1: The Semantic Web Layer Cake[1]

---

[1] http://www.semanticfocus.com/blog/entry/title/introduction-to-the-semantic-web-vision-and-technologies-part-2-foundations/

Figure 1 depicts some of the key concepts and technologies that are important to the Semantic Web. These are the URIs, the RDF Model, the Ontologies and the Query layer.

URI, the Uniform Resource Identifier as defined in RFC 3986[2], is being used to identify resources. The Linked Data best practice recommends using HTTP URIs to identify every resource.

The most important building block among the Semantic Web technologies is the Resource Description Framework (RDF)[3]. RDF is a common framework for describing resources and a standard model for data interchange on the Web. With RDF, one can model information by describing concepts with statements expressed in a simple three-part form called a "triple". A triple is comprised of a subject, predicate, and object, which can be a resource identified by URIs or the object part can be a literal value. RDF is leveraging XML namespaces to provide a method to avoid element name conflicts.

Built on top of RDF, RDF Schema[4] is the first level providing tools to describe classes of entities and their relationships. OWL (Web Ontology Language)[5] builds on top of RDFS and provides tools to formalize ontological relationships and constraints. While RDFS is essential for LOD development, OWL adds further reasoning capabilities on top of the dataset and thus it is preferred.

2.2 SPARQL

SPARQL[6] is a set of specifications that provide languages and protocols to query and manipulate RDF graph content on the Web or in an RDF store. The

---

[2] https://www.ietf.org/rfc/rfc3986.txt

[3] https://www.w3.org/RDF/

[4] https://www.w3.org/TR/rdf-schema/

[5] https://www.w3.org/OWL/

[6] https://www.w3.org/TR/rdf-sparql-query/

typical SPARQL query contains a set of triple patterns called a basic graph pattern. Triple patterns are similar to RDF triples except that each of the subject, predicate and object may be a variable. Such a graph pattern matches a subgraph when RDF terms from that subgraph can be substituted for the variables and the result is an RDF graph equivalent to the matching subgraph. SPARQL has four query forms, which are the following:

2.2.1 SELECT

The SELECT query form returns variables and their matching bindings directly. An example of a simple query, which is listing courses with their respective IDs, is the following (variable names are starting with a question mark):

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?courseID ?courseTitle
WHERE {
  ?courseID rdf:label ?courseTitle .
}
```

2.2.2 CONSTRUCT

A CONSTRUCT query returns an RDF graph constructed by substituting variables in a set of triple templates. An example of a simple query, which is constructing course titles based on the *Dublin Core title* property from the *rdf:labels*, is the following:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
CONSTRUCT {
  ?courseID dc:title  ?courseTitle .
WHERE {
  ?courseID rdf:label ?courseTitle .
}
```

2.2.3 ASK

An ASK query returns a boolean indicating whether a query pattern matches or not. An example of a simple query, which is returning whether there is a course with the title `"Machine Learning 101"`, is the following:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
ASK {
  ?courseID dc:title  "Machine Learning 101" .
}
```

2.2.4 DESCRIBE

A DESCRIBE query returns an RDF graph that describes the resources found. An example of a simple query, which is returning details about a course with the title `"Machine Learning 101"`, is the following:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
DESCRIBE ?courseID WHERE {
  ?courseID dc:title  "Machine Learning 101" .
}
```

2.3 SPARQL UPDATE protocol

SPARQL 1.1 Update is an update language for RDF graphs. It uses a syntax derived from the SPARQL Query Language. Update operations are performed on a collection of graphs in a Graph Store. Operations are provided to create, update and remove RDF graphs in a Graph Store.

2.3.1 INSERT

The INSERT operation adds some triples, given inline in the request, into the Graph Store. An example of a simple operation inserts the following triplet to the triple-store:

Subject: `<http://example/course1>`

Property: `<http://purl.org/dc/elements/1.1/title>`

Object: `"Machine Learning 101"`

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
INSERT DATA
{
  <http://example/course1> dc:title "Machine Learning 101" .
}
```

2.3.2 DELETE

The DELETE operation removes triples, given inline in the request, from the Graph Store.

2.3.3 LOAD/CLEAR

The LOAD operation reads an RDF document from an IRI and inserts its triples into the specified graph in the Graph Store. The CLEAR operation removes all the triples in the specified graph(s) in the Graph Store.

2.4 Ontologies

The term ontology originates from philosophy and deals with theories of being and existence. It was introduced into computer science in the context of Artificial Intelligence research and gained popularity within the field of the Semantic Web.

A widely used definition was given by Thomas R. Gruber:

"*An ontology is a formal specification of a shared conceptualization of a domain of interest.*" [26]

An ontology formally represents the set of concepts within a domain, it describes the classes of the entities, the entities with their properties and the relationships among the classes. A similar concept to ontology is a vocabulary. Vocabularies define the concepts, their relationships, and constraints that are used to describe and represent an area of concern.

In order to create a well-designed Linked Open Dataset, its data model has to be specified, which is composed of the following main tasks: First, the available task specific vocabularies and ontologies should be chosen, they will be specified under the used namespaces. Then the used classes and their relationship should be specified and finally, the possible properties of the entities for each class should be collected. If it is possible, the property should be a member of an existing vocabulary or ontology. If a necessary property does not exist yet, then it should be created for the given data model. For each property, its meaning, its URI, its domain and range should be specified.

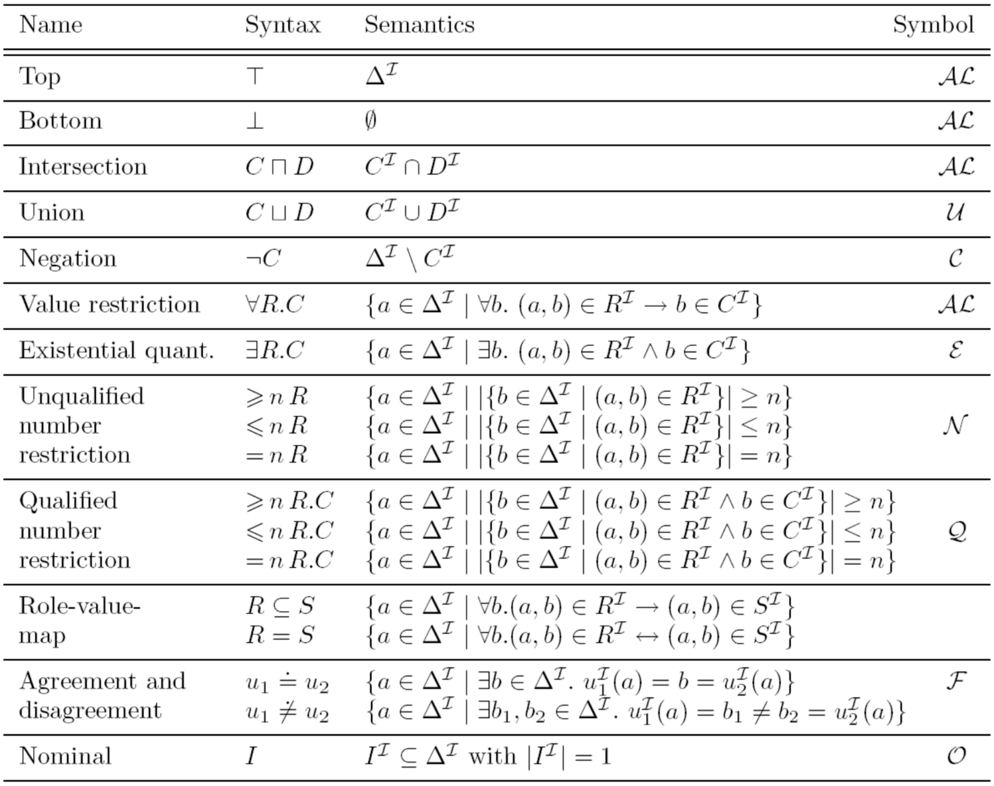
2.4.1 Description logic

Description logic is a family of logic-based knowledge representation languages that can be used to represent the terminological knowledge of an application domain in a structured way [5].

The simple $\mathcal{ALC}$ description logic, which consists of a set $\mathcal{A}$ of atomic concepts, a set $\mathcal{R}$ of roles, and a set $\mathcal{I}$ of individual constants. Sentences, also called as T-Box sentences, are defined as subsumption relations $C1 \sqsubseteq C2$ between concepts, where concepts follow the grammar:

$$C ::= \mathcal{A} \mid \top \mid \bot \mid C1 \sqcup C2 \mid C1 \sqcap C2 \mid \neg C \mid \forall \mathcal{R}.C \mid \exists \mathcal{R}.C$$

Sentences can also be A-Box sentences, which are membership assertions of individuals in concepts (written as $a : C$ for $a \in \mathcal{I}$) or pairs of individuals in roles (written as $\mathcal{R}(a, b)$ for $a, b \in \mathcal{I}$, $\mathcal{R} \in \mathcal{R}$).



Name	Syntax	Semantics	Symbol
Top	$\top$	$\Delta^{\mathcal{I}}$	$\mathcal{AL}$
Bottom	$\bot$	$\emptyset$	$\mathcal{AL}$
Intersection	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	$\mathcal{AL}$
Union	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	$\mathcal{U}$
Negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$	$\mathcal{C}$
Value restriction	$\forall R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \forall b.\ (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$	$\mathcal{AL}$
Existential quant.	$\exists R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b.\ (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$	$\mathcal{E}$
Unqualified number restriction	$\geqslant n\, R$ $\leqslant n\, R$ $= n\, R$	$\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}\rvert \geq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}\rvert \leq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}\rvert = n\}$	$\mathcal{N}$
Qualified number restriction	$\geqslant n\, R.C$ $\leqslant n\, R.C$ $= n\, R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\rvert \geq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\rvert \leq n\}$ $\{a \in \Delta^{\mathcal{I}} \mid \lvert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\rvert = n\}$	$\mathcal{Q}$
Role-value-map	$R \subseteq S$ $R = S$	$\{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in R^{\mathcal{I}} \rightarrow (a,b) \in S^{\mathcal{I}}\}$ $\{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in R^{\mathcal{I}} \leftrightarrow (a,b) \in S^{\mathcal{I}}\}$	
Agreement and disagreement	$u_1 \doteq u_2$ $u_1 \not\doteq u_2$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b \in \Delta^{\mathcal{I}}.\ u_1^{\mathcal{I}}(a) = b = u_2^{\mathcal{I}}(a)\}$ $\{a \in \Delta^{\mathcal{I}} \mid \exists b_1, b_2 \in \Delta^{\mathcal{I}}.\ u_1^{\mathcal{I}}(a) = b_1 \neq b_2 = u_2^{\mathcal{I}}(a)\}$	$\mathcal{F}$
Nominal	I	$I^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ with $\lvert I^{\mathcal{I}}\rvert = 1$	$\mathcal{O}$

Table 1: Description Logic constructs [4]

Table 1 summarizes the different DL constructs and concepts [4]. Concepts (denoted by C and D) are built from role names (denoted by R and S), concept names (also called atomic concepts), the ⊤ top (which is all-inclusive) and ⊥ bottom concepts (represents nothing, an empty set), using the following constructors: intersection (C ⊓ D), union (C ⊔ D), negation (¬C), value restriction (∀R. C), existential restriction (∃R. C) unqualified number restrictions (≥ n R and ≥ n R) and qualified number restrictions (≥ n R.C and ≥ n R.C). In addition to the concept subsumption axioms introduced above, one can also define a hierarchy between roles ($R \sqsubseteq S$). The role hierarchy and transitivity axioms are sometimes referred to as the R-Box (role-box). DL languages can also be extended with datatypes, such as integers or strings.

The logic SROIQ [33], which is the logical core of the Web Ontology Language OWL 2 extends $\mathcal{ALC}$ with the following constructs:

(i) complex role boxes (denoted by $\mathcal{SR}$): these can contain: complex role inclusions such as $\mathcal{R} \circ \mathcal{S} \sqsubseteq \mathcal{S}$ as well as simple role hierarchies such as $\mathcal{R} \sqsubseteq \mathcal{S}$, assertions for symmetric, transitive, reflexive, asymmetric and disjoint roles (called R-Box sentences), as well as the construct $\exists \mathcal{R}.\mathit{Self}$ (collecting the set of *R-reflexive points*);

(ii) nominals (denoted by $\mathcal{O}$);

(iii) inverse roles (denoted by $\mathcal{I}$);

(iv) qualified and unqualified number restrictions (denoted by $\mathcal{Q}$).

2.5 Linked Data

Linked Data is a practical application of the Semantic Web technologies for connecting data worldwide. The Open Data pursuit has achieved remarkable progress in Europe as well, and studies have shown that it has a positive impact on the quality of education at the university level too.

Since the definition of Linked Open Data (LOD) in 2006, LOD datasets are appearing all around the world rapidly, expanding the way that we access data. Different institutes and data owners are figuring out new ways to export their data in LOD triples, integrate external linked datasets into their collections, and develop new interfaces.

The principles of Linked Open Data are related to publishing and interlinking structured data on the Web so that computers can read it automatically. This approach enables data from multiple different sources to be connected and queried. The Linked Data concept – invented by Tim Berners-Lee in 2006 – is based on the following four principles [7]:

1. *Use URIs as names for things*
2. *Use HTTP URIs, so that people can look up those names*
3. *When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)*
4. *Include links to other URIs, so that they can discover more things*

The publication of Linked Data is based on standard Web technologies such as Uniform Resource Identifiers (URIs), the Hypertext Transfer Protocol (HTTP) and the Resource Description Framework (RDF).

The 4th recommendation ensures there are links within the different datasets. The standard data model for Linked Open Data is the Resource Description Framework (RDF). Linked Data builds links between arbitrary things described in RDF.

Figure 2 shows datasets that have been published in Linked Data format, by contributors to the Linking Open Data community project and other individuals and organizations. It is based on metadata collected and curated by contributors to the Data Hub[7]. It consists of seven major and multiple smaller domains linked together, mostly through major hubs, like DBpedia[8]. One domain where Linked Data plays a major role is life sciences, almost one-third of all open datasets are related to this domain.

---

[7] http://datahub.io/

[8] http://wiki.dbpedia.org

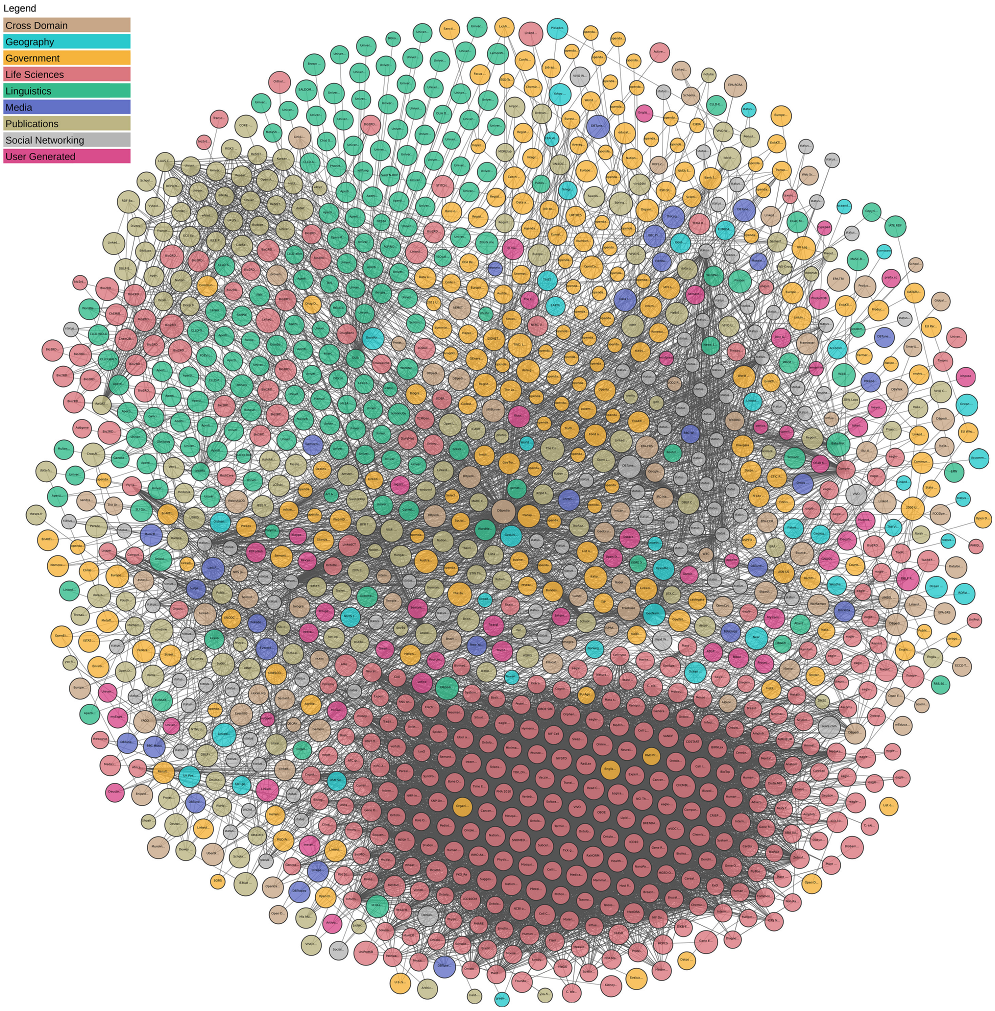


Figure 2: The size of the Linked Open Data cloud in 2018[9]

---

[9] https://lod-cloud.net

3 Linked Data Enrichment with Self-Unfolding URIs

Publishing Linked Data on Linked Data Platform [67] demands certain best practices, e.g. to provide meaningful URIs to identify entities. For human usage – especially for manual entry – it is important to encode some semantics in the URI structure. This can cause redundancy in the data, as these entities should be linked to other entities while this information may present in the URI as well. If this redundancy cannot be avoided, some automated mechanism should take care of the maintenance or verification.

In the context of Linked Data usage in order to enable correct queries on the data, it is often crucial to have a complete description of the entities as one might expect certain properties for certain entity types. For this purpose, we propose the Self-Unfolding Semantic URIs: these are URIs following a specific pattern and a template, which describes the structure the entity should have. Let's suppose a LOD dataset is given and there is a mechanism (e.g. a trigger) monitoring the data that in case of a special type of entity appearance generates new LOD triples that are semantically derived from the old ones.

Automatic Linked Data expansion can be categorized by

(i) the characteristics of the data that triggers the automatic data generation,

(ii) the method of the generation,

(iii) the structure of the new triples and

(iv) the mechanism supporting the automatism.

The process is indispensable for efficient data management of certain types of LOD datasets. For example, a proper, re-usable OWL-Time Interval description requires at least 7-8 triples, which is quite tedious and error-prone for manual input. Thus, in a temporal dataset, the automatic generation of the time Instant description RDF triples is inevitable.

3.1 Linked Data Enrichment

Linked Data Enrichment is the process leveraging implicit or hidden semantics built into a dataset and made explicit by RDF tools. Previous studies in this field focused mainly on link generation to semantically related resources, which in most cases means the automatic insertion of *owl:sameAs* statements between instances of different resources, (e.g. to DBpedia or to Geonames[10] datasets) [30] or between the corresponding concepts (e.g. ontology classes or properties) [49]. The latter is the main objective of ontology alignment [21] and schema alignment [52]. Answering complex queries often requires accessing and combining information from multiple datasets, which can be achieved by federated query processing. Different approaches to federated query processing over linked data are analyzed in [28]. The authors study how different design alternatives affect the performance and practicality of query processing and define a benchmark for federated query processing, comprising a selection of data sources in various domains and representative queries.

Semantic Sensor Web is the combination of sensor and Semantic Web technologies, where the encoding of sensor descriptions and observation data with Semantic Web languages enables more expressive representation, advanced access, and formal analysis of sensor information. Dynamic enrichment is a current research topic in the field of linked sensor data streams.

---

[10] http://www.geonames.org

In [29] authors propose the use of a Complex Event Processing engine with a dynamic enrichment component that expands the sensor information items before evaluating them. The authors suggest a prototype to realize situation awareness over large-scale and open web sensor networks. For the core processing model of the enrichment, a relational query model is used.

Transforming sensor-based data into RDF and making it available using HTTP requires the use of sensor data related URIs. In [58] authors propose a URI-based mechanism to identify and access Sensor Data coming from sensor networks and they propose several URI designs to represent Time, Space and sensor identity information in URIs. In [11] the most relevant challenges of the Semantic Sensor Web are described, where the integration and fusion of data coming from different sensor networks (with varying qualities of service and different throughput rates, geographical scales) and other sources (e.g. static data or archived sensor data) are emphasized.

Correndo, et al. [12] address the issue of representing time entities (i.e. Instants and Intervals) as Linked Data, and how to exploit topological temporal relationships in order to increase the connectivity degree within Linked Data sets. They present an approach to describe temporal entities as reusable URIs that can be adopted by data publishers as a temporal context to their information resources. The approach identifies a set of discrete temporal entities as relevant for a certain domain (e.g. financial years for the public sector) while a RESTful API is provided to users to dynamically create their own temporal entities. Once a dynamic temporal URI is resolved, information is provided to situate such URI in reference to the domain relevant entities. The URI resolution employs simple topological temporal reasoning in order to exploit the qualitative relationships between entities.

3.2 Use cases

In the following, the description of some use cases is given in order to highlight the utility of the suggested method. We have identified a temporal and a spatial use case and a third one combining these.

3.2.1 OWL Time entities

There are multiple ways to model temporal information, but one of the mostly used ontology for this purpose is the OWL Time Ontology[11]. It provides basic constructs to define and describe points and intervals bounded with a start and endpoint in the temporal space. OWL Time provides two approaches to describe a point of time: either using the xsd:datetime datatype or using the DateTimeDescription class. While the first one offers an easy way to define a point of time by a well-structured string, it lacks some of the features the DateTimeDescription class provides: it uses seconds as the default unit type and requires a full date and time described. On the other hand, manually modeling and maintaining DateTimeDescription entities are error-prone and tiring, because they require at least 7-8 triples in a format that is really re-usable in a semantic sense. For example, defining a time interval entity requires the following triples:

```
ex:meetingInterval
   a                    time:Interval ;
   time:hasBeginning    ex:meetingStart ;
   time:hasDurationDescription ex:meetingDuration .

ex:meetingStart
   a                    time:Instant ;
   time:inDateTime      ex:meetingStartDescription.
```

---

[11] http://www.w3.org/TR/owl-time/

```
ex:meetingStartDescription
   a                    time:DateTimeDescription ;
   time:dayOfWeek       time:Monday;
   time:day             8 ;
   time:hour            8 ;
   time:minute          0 ;
   time:month           9 ;
   time:unitType        time:unitMinute;
   time:year            2014 .

ex:meetingDuration
   a                    time:DurationDescription ;
   time:minutes         90 .
```

If a given LOD dataset contains a lot of temporal information, manually publishing all the necessary triplets would be cumbersome. We suggest instead to encode the necessary information in the URI of the time entity and use the self-unfolding mechanism to auto-generate the required triplets based on the encoded information in the URI.

In the above example the URI of the interval entity should contain all the data that is needed to generate the corresponding triples, for example, the following is a good choice:

```
<http://example.org/data/Interval;year=2014;month=9;day=8;hour
=8;minute=0;durationHour=1;durationMinute=30>
```

3.2.2 Spatial entities

Using Spatial Entities to specify locations for events or files (like an image was created on a specific location) is a handy way to express geo-location. Geo-location can be easily captured and represented by coordinates, so encoding latitude and longitude in the URI identifying the geo-location is a plausible method. Additional information can be generated by connecting related entities with the coordinates, like the country, city, region, landmark nearby found in a geodatabase, like Geonames. With federated SPARQL Construct queries, such enrichment can be scheduled or triggered by the advent of a new instance of a geo-location.

The enriched geo-location data for the URI

```
<http://example.org/Point;lat=47.158775;long=18.88149>
```

could be the following:

```
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<http://example.org/point;lat=47.158775;long=18.88149>
  a                geo:Point ;
  geo:lat          "47.158775" ;
  geo:long         "18.88149" ;
  foaf:based_near  <http://sws.geonames.org/3048446/> .
```

3.2.3 Measurement representation from Semantic Sensor Networks

A Semantic Sensor Network publishes measurement data in a predefined cadence or an ad-hoc manner. Such data might include the location of the measurement, the exact time, the measured value and the sensor ID. Each measurement event is identified by a URI and represented by a set of triplets. In [58] authors suggest models for URI structures to identify and access Stream Data coming from sensor networks. URI schema is proposed containing the sensor identifier, the time of the measurement and the space information. We go further and suggest putting the value of the measurement also in the URI scheme. In this use case, the URI containing the sensor ID, the time, space and the measurement information would be sent to the data processing center, where the data enrichment would take place. The data enrichment would use a template containing extra information in RDF triplets about the sensor geographical features, sensor type, measurement interpretations, and references. From the information encoded in the URI with the help of the template, the raw results of the measurement and also some interpretations and references of the measurement would be generated. This way it becomes simple to query out-of-normal-range events.

The following (highly simplified) example shows the result of the generation of sensor observation RDF data from the URI defining the measurement:

```
<http://example.org/observation;instant=2015-01-03T10-38-
43;lat=60.158775;long=24.88149;value=12.2>
```

The auto-generated triplets could be (in Turtle):

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dul: <http://www.loa-cnr.it/ontologies/DUL.owl#> .
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#> .

<http://example.org/observation;instant=2015-01-03T10-38-
43;lat=60.158775;long=24.88149;value=12.2>
        a ssn:Observation ;
        ssn:observationResult "12.2" ;
        ssn:observationResultTime
<http://example.org/instant;instant=2015-01-03T10-38-43> ;
        dul:hasLocation
<http://example.org/point;lat=60.158775;long=24.88149> .
```

3.3 Generic scenarios

In the following generic LOD publishing scenarios are described, where the automatic Linked Data enrichment has been proven useful. We have identified four scenarios, i.e. four problem types, where automatic data generation with self-unfolding URI could provide a solution. These are entities with composite keys in URI, Blank Node (Anonymous Resources) Candidates, linking related entities based on URI and reasoning on information encoded in the URI.

3.3.1 Modeling composite keys in URI

Modeling entities without a strong primary key candidate in RDF has its own challenges. Composite keys identify such entities more naturally, however defining a good and representative URI with all keys included will cause redundancy in the model, since the keys will be encoded in the entities URI syntactically and also as triplets semantically.

Our proposal to resolve this dilemma is automatic triplet generation based on URI patterns and templates, which provides a mechanism to maintain the redundancy of information encoded in a URI and paraphrased by triplets.

3.3.2 Generating Blank Node entities from URIs

In some cases, the URI of an RDF entity is not important or not used, for this reason, the concept of Blank Nodes was introduced. However, this construct does not exist in OWL, thus cannot be used in tools like Protégé supporting OWL. Another drawback of using Blank Nodes is the locality of their scope. Manually creating and maintaining URIs for otherwise Anonymous Resources is costly and error-prone as usually this creates redundancy: one needs to encode some semantics in the URI as well as describing the instance with RDF constructs.

Our proposal is to encode the necessary information in the URI of the Blank Node entity thus it becomes a named entity and to use automatic triplet generation based on URI patterns and templates.

3.3.3 Linking related entities based on URI

Linked Data Enrichment with SPARQL Construct queries is capable to act as a reasoner, with the main role of making implicit relationships explicit by creating the corresponding triplets. For example, it is able to link high-level and low-level temporal entities as overlapping or succeeding. Linking related events (e.g.: Semesters to Lectures) can be inferred by simple attribute calculations too.

3.3.4 Reasoning on information encoded in URI

In [50] Polleres, Axel, et al. describe methods to implement attribute equations which allow a very restricted form of simple numerical equations in multiple variables. These methods are using SWRL rules and SPARQL Query rewrite. Additionally, SPARQL Construct queries provide an expressive way to implement information extraction encoded in URI.

3.4 Implementation

In this section, the functional requirements are described: the URI pattern definition, the corresponding RDF templates and the structure of the SPARQL Insert queries achieving the data generation.

Closely tied to the publication of Linked Data is the specification of a standard read/write interface, which is the goal of the Linked Data Platform [60]. Such a platform could provide a natural place to implement the self-unfolding URIs: one could POST a triplet to an LDP Indirect Container (LDP-IC) and the platform unfolds the entity.

3.4.1 URI Patterns

As the SPARQL language has limited expressiveness, it is important to design the structure of the URIs in a way that supports the processing by SPARQL Inserts and also conforms to other best practices. In [19] there are 8 URI design patterns describing how to assign identifiers within a dataset, where all 8 patterns are widely used and tested in the field. It should be noted that none of them is a good candidate to apply for composite key modeling. From the 8 URI design patterns, the closest one to our need is the Hierarchical URI pattern, which can be applied when a natural hierarchy exists within the set of resources (e.g. /books/001/chapters/1). However, in the case of composite key modeling, the concept hierarchy does not always exist, i.e. the parts of the key are often in a coequal relationship.

Because of this, we propose the use of Matrix URI pattern[12] as it can use multiple independent parameters of the entity. A URI following this pattern starts with a base part, followed by the type of the resource and then optionally each part of the composite key:

```
<Base URI>/<type>[;<Property>=<Value>]
```

3.4.2 Conceptual RDF Template

The purpose of using templates in the self-unfolding URI process is to express rules, which determine the generation of the new information based on the entity's URI. There is a constant challenge for Linked Data consumers that the noise in the data can lead to fuzzy data structures too. The aim of defining a template for a given type of entity is to support the query consistency. Each template implementation has to define the rules to extract information from the URI and optionally rules to infer any implicit relations or attribute values. The template provides a URI pattern and prescribes RDF triplets to be generated based on the values extracted from the URI. Following is the high-level structure of such a template:

```
<URI> rdf:type  <type with namespace>
      [; <property> <value>|<related URI>] .
```

Related entities can be generated recursively applying the same template. An example of a template generating an OWL Time Instant entity can be the following:

---

[12] http://www.w3.org/DesignIssues/MatrixURIs.html

```
<http://example.org/Instant;year={year};month={month};day={day
};hour={hour};minute={minute}> =====>
<http://example.org/Instant;year={year};month={month};day={day
};hour={hour};minute={minute}>
 rdf:type time:Instant;
 time:inDateTime.
<http://example.org/DateTimeDescription;
year={year};month={month};day={day};hour={hour};minute={minute
}> .
<http://example.org/DateTimeDescription;year={year};
month={month};day={day};hour={hour};minute={minute}>
 rdf:type time:DateTimeDescription ;
 time:hour "{hour}"^^xsd:nonNegativeInteger ;
 time:minute "{minute}"^^xsd:nonNegativeInteger;
 time:month "{month}"^^xsd:nonNegativeInteger ;
 time:day "{day}"^^xsd:nonNegativeInteger ;
 time:year "{year}"^^xsd:nonNegativeInteger .
```

3.4.3 Implementation with SPARQL Insert statements

SPARQL Insert operation can be utilized to implement templates defined earlier. For each entity having a URI matching the URI pattern a SPARQL Insert statement produces the corresponding properties and objects or attribute values. These update queries are the implementation of the templates describing the entity.

Atomic operations include the following events: an entity with a new URI has been created or an entity has been deleted. URI changes are captured as a delete and a create operation. If a generated entity matches a pattern again, we are talking about chained entity generation.

The following example is about generating a (highly simplified) sensor observation entity as described in the section 3.1.3 with time and location information:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX dul: <http://www.loa-cnr.it/ontologies/DUL.owl#>
INSERT {
   ?uri     rdf:type ssn:Observation ;
   ssn:observationResultTime ?instant_uri ;
   dul:hasLocation ?loc_uri ;
   ssn:observationResult ?value .
}
WHERE {
   ?uri ?p ?o.
   FILTER(STRSTARTS(STR(?uri),
"http://example.org/observation"))
   BIND(IRI(REPLACE(REPLACE(REPLACE(str(?uri),"observation",
"instant"),'(value=[^;]*;?)|(lat=[^;]*;?)|(long=[^;]*;?)',
""),';$', "")) AS ?instant_uri) .
   BIND(IRI(REPLACE(REPLACE(REPLACE(str(?uri),"observation",
"point"),'(value=[^;]*;?)|(instant=[^;]*;?)', ""),';$', "")) AS
?loc_uri) .
   BIND(REPLACE(REPLACE(str(?uri), '^.*value=', "") , ';.*$',
"") AS ?value) .
}
```

The following URI

```
<http://example.org/observation;instant=2015-01-03T10-38-
43;lat=60.158775;long=24.88149;value=12.2>
```

will trigger to generate the following (in Turtle):

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dul: <http://www.loa-cnr.it/ontologies/DUL.owl#> .
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#> .
<http://example.org/observation;instant=2015-01-03T10-38-
43;lat=60.158775;long=24.88149;value=12.2>
```

```
        a                      ssn:Observation ;
        ssn:observationResult  "12.2" ;
        ssn:observationResultTime
<http://example.org/instant;instant=2015-01-03T10-38-43> ;
        dul:hasLocation
<http://example.org/point;lat=60.158775;long=24.88149> .
```

The Instant entity

```
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix time:  <http://www.w3.org/2006/time#> .
<http://example.org/instant;instant=2015-01-03T10-38-43>
        a                   time:Instant ;
        time:inXSDDateTime  "2015-01-03T10-38-43" .
```

is generated by the following construct:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX time: <http://www.w3.org/2006/time#>
INSERT {
?uri rdf:type  time:Instant;
     time:inXSDDateTime ?instant.
}
WHERE {
   ?uri ?p ?o.
   FILTER(STRSTARTS(STR(?uri), "http://example.org/instant"))
   BIND(REPLACE(REPLACE(str(?uri), '^.*instant=', "") , ';.*$',
"") AS ?instant) .
}
```

The third example demonstrates how the location entity can be generated. With a federated query to the Factforge SPARQL endpoint[13] we are enriching the result with nearby entity:

---

[13] http://factforge.net/sparql

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX omgeo: <http://www.ontotext.com/owlim/geo#>
INSERT {
?uri  rdf:type geo:Point;
      geo:lat ?lat;
      geo:long ?long;
      foaf:based_near ?location.
}
WHERE {
   ?uri ?p ?o.
   FILTER(STRSTARTS(STR(?uri), "http://example.org/point"))
   BIND(REPLACE(REPLACE(str(?uri), '^.*lat=', "") , ';.*$', "")
AS ?lat) .
   BIND(REPLACE(REPLACE(str(?uri),  '^.*long=',  "")  ,  ';.*$',
"") AS ?long) .
   SERVICE <http://factforge.net/sparql> {
    SELECT ?location
    {
     ?location omgeo:nearby(?lat ?long "1km").
    }
    LIMIT 1
   }
}
```

The enriched geo-location is the following:

```
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
<http://example.org/point;lat=47.158775;long=18.88149>
        a        geo:Point ;
        geo:lat  "47.158775" ;
        geo:long "18.88149" ;
        foaf:based_near <http://sws.geonames.org/3048446/> .
```

3.4.4 Maintaining Consistency

It is advised to maintain meta information about any auto-generated data, in order to keep it in sync with the manually created part. It can be a good practice to maintain these in a separate graph, or in the case of LDP in the LDP Indirect Container so all data can be viewed with or without the generated information.

One drawback of this method is the potential inconsistency during the (slight) delay between an entity was created and unfolded or deleted and the remaining orphan related entities were removed. An entity is orphan if there is no triplet where its URI is on the right side: `<?s ?p {Orphan URI}>` To completely remove an entity represented by a URI pattern defined in Section 3.4.1, all triplets containing the corresponding URI need to be deleted. Altering or removing the original entity from the triple store needs to be reflected by the related entities too, a garbage collection algorithm can look after the related orphan entities. If an auto-generated entity is orphan, it means the parent entity it was generated from no longer exists, so it can be deleted as well as all the sibling entities recursively, as shown in this example:

```
DELETE { ?orphan ?p ?v }
WHERE
 { ?orphan ?p1 ?v .
   OPTIONAL { ?parent ?p2 ?orphan. }
   FILTER (!BOUND(?parent))
   FILTER(STRSTARTS(STR(?orphan), "http://example.org/point"))
 }
```

3.5 Conclusion

We think that the proposed URI structure may have a wider use as a Linked Data pattern [19] since there is no other pattern at the moment to deal with the specific scenarios described in Section 3.2. Applying the self-unfolding URIs eases the authoring and maintenance of Linked Data especially in cases when simple, frequently occurring concepts (such as date intervals) need to be represented in a rather complex way. The suggested mechanism also provides an elegant way to generate standard URIs for frequent entity types.

Our proof of concept implementation was based on a scheduled operation (e.g. once a day) for the recurring entity generation. While this proved to be a simple solution, there is a period of time when not all entities are unfolded thus leading to data consistency issues.

SPARQL was used for the purpose of demonstration, any other – more expressive – technology can be used to define the unfolding logic. The above SPARQL examples also demonstrating the complexity of URI and string operations within a query. In a future SPARQL version, this can be a consideration for additional language elements making the string operation more effective within queries.

It is advised to maintain meta information about any auto-generated data, in order to keep it in sync with the manually created part. It is a good practice to maintain these in a separate graph, or in the case of Linked Data Platform (LDP) in the LDP Indirect Container so all data can be viewed with or without the generated information. Future work should include implementing a trigger type mechanism to capture changes within a triple store and to reduce the inconsistent period to a more acceptable one.

Closely tied to the publication of Linked Data is the specification of a standard read/write interface, which is the goal of the Linked Data Platform. Such a platform could provide a natural place to implement the self-unfolding URI mechanism. In LDP a triplet would be sent with POST to an LDP Indirect Container (LDP-IC) and the platform could unfold the entity.

4 The OLOUD Ontology

The Ontology for Linked Open University Data (OLOUD) is a practical approach to model the education activities of a typical Hungarian university. To facilitate the implementation of Smart Universities, OLOUD aims to support data integration from multiple data sources and provide personal timetables, course-, subject-, curriculum and other type of information for students and lecturers.

4.1 Introduction

The main motivation of developing a new ontology born during our initial work of publishing university related Linked Open Data: In the domain of university courses there are existing ontologies and our initial approach was to build a linked open university dataset using these ontologies. It turned out however, that the existing ontologies did not provide full coverage to the requirements and some of them even contained inconsistencies. Facts about entities related to the course-subject-curriculum concepts and connections between them cannot be described by the existing ontologies in the required level of detail. This recognition confirmed the need of developing a new ontology.

We chose the ontological representation of the new model, as it is the most modern formal description method for a problem domain. Ontologies can be used as a data model to influence or integrate traditional SQL databases or as an RDF schema in triple stores and even in reasoners or rule systems to enhance the collected data. The original objective was to develop a generic data model for university ‘course related’ data. During the work, we noticed that although the Bologna Process[14] ensures a certain level of compatibility for education systems in the EU, this does not reach to deeper constructs of the educational model. We found that especially the meanings of *course*, *subject* and *study program* are quite different in the available educational models in Europe. Therefore, we decided to limit the scope of our work to the Hungarian interpretation of the concepts. During the ontology development, we relied on the existing concept definitions and data structures used by university information systems in Hungary such as the Neptun student information system[15] or in the widely used Moodle e-learning platform[16].

Presenting 'course related' information requires data from multiple information systems at a typical university. As these systems are usually not

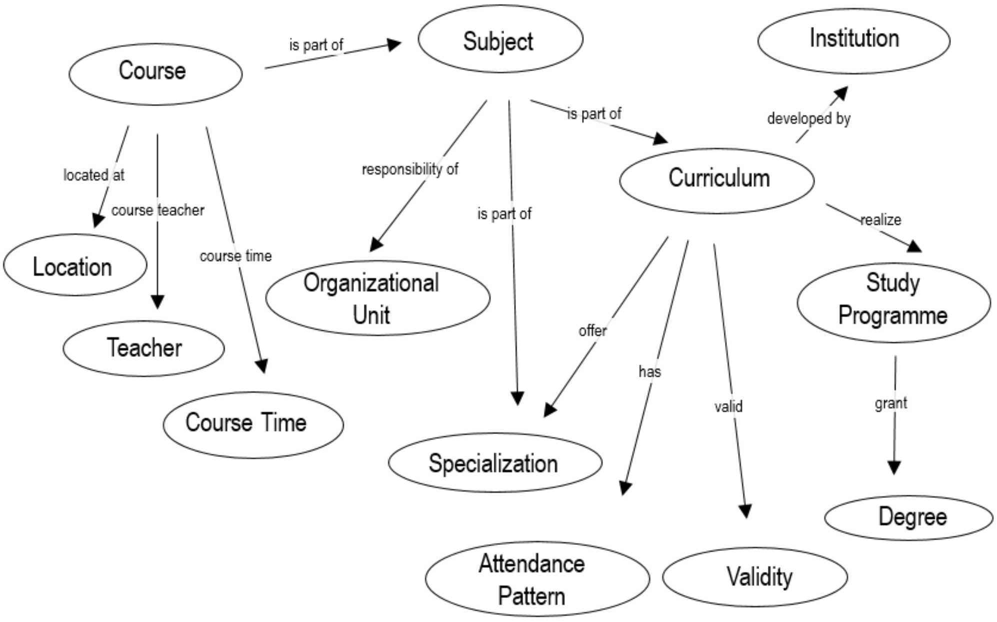


Figure 3: Major concepts in the Hungarian higher education

---

[14] https://ec.europa.eu/education/policies/higher-education/bologna-process-and-european-higher-education-area_en

[15] https://neptun.uni-obuda.hu/

[16] https://elearning.uni-obuda.hu/

fully integrated and the access to the data is limited, significant effort is necessary to successfully navigate through the potential obstacles.

Foreign students unfamiliar with local specifics might find the process even more challenging. With our data model, we want to support automatic collection and management of integrated university ‘course related’ data and future mobile and web applications, that are built on the top of this data. In the following the major concepts and use cases are defined, that we aim to support with our Linked Data approach.

In the case of Hungarian universities, it is crucial to understand the difference between Subjects and Courses. Figure 3 summarizes the major concepts for university students and teachers concerning university courses, subjects, curricula and study programme. Course is the elementary unit of the educational process, where date, location, teacher and students are assigned. Course is the framework, which has a specific training type (like lecture, practice or seminar) and has some requirements that students must complete. Courses are organized in individual sessions in a weekly or a custom cadence within a semester, and such a course event is called the session of the course. Subject is a higher-level component of the training process, it is the unit of the curriculum with a specified training content, and fulfillment is rewarded with a number of credits. It may include more courses, which all must be completed to complete the subject. Curriculum describes all the compulsory and optional subjects one has to complete in order to realize a study program and finally to get a certain degree. For each curriculum there are several specializations, each providing multiple attendance patterns (e.g. full-time and part-time) and each curriculum has a validity as universities are renewing these occasionally.

4.2 Linked Data consumption use cases

One of the main objectives of OLOUD is to enable universities creating and publishing Linked Data. In the following we list some of the tasks we aim to support.

University courses are organized into a series of lectures, lab exercises or seminars, either in a weekly or in a custom cadence within a semester. There might be multiple labs advertised for a course, so students can choose the most suitable to their preferences. This creates the challenge of assembling a personal timetable for students and lecturers avoiding conflicts and considering personal preferences and requirements. Students would benefit from an aggregated view of course descriptions (title, identifier, abstract, and dependencies), time and location. A personal information service may provide students with on-demand information about their daily schedule, step-by-step navigation to the next lecture, overlaps of classes, etc.

Other use case is the longer-term planning: what courses a student should attend to fullfil a specific curriculum. Usually there are no timetables given at the Hungarian universities, just a list of courses to be completed, and a dependency graph among the courses, which defines the order for each. Some courses are advertised in every second semester only. Some universities recommend a specific order of courses but following such an order breaks easily if for example a single course is not completed in the suggested semester. Thus, students face a kind of constraint-satisfication problem to solve every time a semester starts. For this purpose, students need a personal advisor, recommending the best way for them to fulfill the curriculum requirements. This advisor needs to consider where the students are on their roadmap, what courses they should focus on, what are the personal preferences (e.g. preferred number of courses or credits per semester and preferred specializations) and lastly what courses are being advertised.

The last example use case is an opportunity to use data for originally unconsidered purposes. University resources (rooms, equipment) are used by multiple faculties. They can be booked for regular courses, exams in the exam period and other events. Different types of events may have separate registries, thus it is challenging to have an overall view of anticipated resource usage. One needs an overall list of reservations by the reserving person, location and date at least. Using the course related events data, one can easly get a good picture of resource reservation.

According to the above use cases we have defined a set of competency questions, to have a first pass validation on the ontology requirements. With the data utilizing the ontology one should be able to answer the following questions:

- What are the attributes of a given curriculum, like start of validity, end of validity, study programme, degree, language, attendance pattern?
- What type of specializations exist in each curriculum?
- What are the mandatory subjects of a given curriculum?
- What are the mandatory subjects of a specialization?
- What are the attributes of a specific subject, like who is the teacher responsible for, credit number, degree, language, attendance pattern?
- What type of courses consists of a subject?
- What are the courses for a given subject in a specific semester?
- What are the prerequisite subjects of a certain subject?
- What are the attributes of a specific course, like teacher, course type, requirement type, hours?

- On which day a given course is, what time does it start and how long it is?
- How to manage exceptions in a course schedule cadence?
- What is the location of a certain course?
- What courses are partially or completely overlapping in time?
- What is the course schedule (with course identifier, time and lecturer) for a specific lecture hall or lab?
- What is the step-by-step navigation route between two course locations?
- What are the dates of the individual sessions of a given course?
- What event is going to be at a specific location at a specific time?

In the following we use the above formulated questions as competency questions for the evaluation of the ontology.

4.3 Ontologies for education

The scope of this section is to review existing ontologies that are available to describe various aspects of educational courses and to evaluate whether they can be used to model 'course related' information in the Hungarian higher educational landscape. For simplicity, in the following we refer to ontologies, vocabularies and lighter schema constructs as ontology uniformly.

Linked Universities[17] and Linked Education[18] are two European initiatives created to enable education with the power of Linked Data. Linked Universities is an alliance of European universities engaged in exposing their public data as linked data. It promotes a set of vocabularies describing

---

[17] http://linkeduniversities.org

[18] http://linkededucation.org

‘academic related’ entities. LinkedEducation.org is an open platform aimed at further promoting the use of Linked Data for educational purposes.

The Open University in the UK was the first university that created a linked data platform to expose information from its departments. The evolution process of the Open University Linked Open Data platform is described in [16]. This process started as a research experiment and evolved to a data hub for the open content of the university. The platform is now the key information service at the Open University, with several applications and websites exploiting linked data through data.open.ac.uk and establishing connections with other educational institutions and information providers. In the publications, the authors describe the main milestones and tasks accomplished to achieve this state. The Open University datasets can be classified into the following six groups: open educational resources, scientific production, social media, organizational data, research project output, and publication metadata.

The main difference to the Óbuda University is the lack of navigation and timetable data at the Open University. There are 125 classes and 785 properties from 57 public vocabularies to describe the data at the Open University. Their main effort in the modelling was to reuse the most matching terms from existing vocabularies directly instead of being restricted to the semantics of only a few widely-used ontologies. The large number of the used vocabularies, the redundancy in the data and in the used properties are the consequences of their approach.

The general process for building linked open university data and a use case at Tsinghua University are described in [40]. Procedures like choosing datasets and vocabularies, collecting and processing data, converting data into RDF and interlinking datasets are studied. The datasets unfortunately are not available through a public SPARQL endpoint.

The Lucero project analyzed open educational datasets in 2012 [14]. Linked Open Datasets in four universities and four broader educational projects were studied and the most commonly used vocabularies, classes and properties were described. In this case no representations for course, semester or lecture room concepts were found.

Ontology	**Course**	**Subject**	**Curricula, Study Program**	**Speciali-zation**	**Degree**	**Teacher**	**Organi-zation**	**Time**	**Loca-tion**
FOAF						✓	✓		
Vcard						✓	✓		
Event	✓							✓	
W3 Time								✓	
iLoc									✓
Aiiso +	✓	✓	✓	✓		✓	✓		
Teach	✓		✓	✓	✓	✓		✓	✓
XCRI			✓		✓		✓		
Course-Ware	✓								
VIVO	✓			✓		✓		✓	
MLO, ECIM	✓	✓			✓		✓	✓	
Bowlogna	✓		✓	✓	✓	✓	✓		

Table 2: Concept coverage of relevant ontologies

The state of linked data for education is studied in [13]. They collect existing datasets explicitly related to the education field, extract key information, and analyze them. The goal is to better understand what is already available to application developers in this area, what common practices are being used and how the considered datasets connect with each other through common content and vocabulary reuse. They found 144 different vocabularies used in ‘education related’ datasets. The most popular vocabularies are not specific to education, but are used to represent general concepts and relations, such as resource metadata (Dublin Core [17]), people (FOAF [9]), topics (SKOS [45]), time (W3C Time Ontology [34]) and bibliography (BIBO [15]). More education specific vocabularies are also widely used, such as the Academic Institution Internal Structure Ontology (AIISO [62]), or the Model of Learning Opportunities (MLO [46]).

We found a very useful review of vocabularies and ontologies for modelling course information in higher education [6]. Table 2 provides an overview about ontologies related to our use case and their description capabilities. There are big differences in the interpretation and in the elaboration of the used terms in the various ontologies.

AIISO (Academic Institution Internal Structure Ontology) provides classes and properties to describe the structure of an academic institution. It is designed to be used in conjunction with the Participation ontology [63], which stands for describing the roles, that people play within groups. Participation has only one class, but any domain can extend it by creating subclasses for their own roles within their areas of expertise. AIISO Roles [62] is An example of such extension; it describes roles that people play in an academic institution. The AIISO ontology proved to be useful in our work, as it distinguishes at class level the Course and the Subject concepts. These classes are subclasses of KnowledgeGrouping. There is a note in AIISO that this class became deprecated. Probably it was a plan of the authors, but there is not any information on how and when this would be done. AIISO offers only a few properties to describe courses (i.e. code, description, teaches and responsibility) and is mainly used to connect subjects and courses with the organizational structure of the university.

TEACH [38] is a lightweight vocabulary providing detailed properties to describe a course, but it does not model the provider of the course. The concepts in TEACH lack some important features, that are essential for our purposes. For example, the concept 'Subject' is necessary to describe university courses, which does not exist in TEACH. Another problem was, that one would expect an *owl:DataTypeProperty* based on the example data of TEACH, but the ontology itself declared the specific property (e.g. teach:courseDescription, teach:ects) as *owl:ObjectProperty*. These shortcomings were fixed, and the corrected version[19] of the TEACH ontology was used in the first phase of our work. In TEACH one can find further issues though. For example, the properties hasAssignment, hasAssignmentMaterial, and hasCourseMaterial have appeared in the index of terms of TEACH, however they are not defined in the vocabulary specification. The classes Student and Lecture are defined, but there is no property available in the ontology to relate them to a Course. A potential disadvantage of the TEACH ontology is that it is not linked into other ontologies and some of the definitions are missing or do not have domains or ranges specified. Because of the above problems of this vocabulary, we decided not to use it.

XCRI-CAP [61] is the abbreviation for eXchanging Course Related Information, Course Advertising Profile. The term *course* in the UK is equivalent with the term *study programme* in Hungary, thus XRI-CAP does not contain description about course and subject in our terminology. XCRI-CAP is the UK standard for describing study programme marketing information. XCRI represents a lot of data about the provider and the programme and it also differentiates between a programme and the particular presentation of it. XCRI-CAP is in XML format and does not exist in RDF.

---

[19] http://lod.nik.uni-obuda.hu/teach-fixed.owl

The ReSIST Courseware Ontology [54] is a simple ontology with only four classes and many properties like title, teacher, credits, prerequisites, assessment method, etc. It was developed within the ReSIST project between 2006 and 2009. It is an early ontology without any usage in current days. The main problem with this ontology is, that it is closely related to the Aktors ontology, which is no longer defined anywhere online.

The Metadata for Learning Opportunities (MLO) Advertising ontology is similar in a way to XCRI-CAP, because its purpose is to standardize the specifications for describing and exchanging information about learning opportunities. It can be considered the European equivalent of the British Standard XCRI-CAP for advertising learning opportunities. MLO-Adv contains the following four classes:

1. Learning Opportunity Object: an abstract resource used within the context of education or training. It has the following three subclasses:
2. Learning Opportunity Provider: a person or organization that offers the learning opportunities
3. Learning Opportunity Specification: description of a learning opportunity, consisting of information that will be consistent across multiple instances of the learning opportunity.
4. Learning Opportunity Instance: single occurrence of a learning opportunity, it might have a particular date or location.

MLO includes some properties from Dublin Core Elements such as contributor, date, description, identifier, subject, title, and type. ECIM [10] is an extension of MLO, which provides a common format for representing credits awarded for completion of a learning opportunity. XCRI, MLO and ECIM ontologies are similar in that they differentiate between a course specification and a course instance or course offering. The specification contains information about a course or a study programme that remain consistent from one session to the next, whereas the instance defines those aspects that vary between sessions for example location or start date. This has the advantage that there will be a smaller amount of data that needs update between years and offerings.

The goal of the VIVO ontology [8] is to represent academic research communities, and thus it enables the discovery of researcher interests, activities, and accomplishments. In a later phase of our work VIVO may be useful to represent research groups within the university, including researchers' grants and external roles. Currently, its focus is quite different from the focus of OLOUD, for example VIVO has its own Course class, but its main properties are credits and prerequisites.

The Bowlogna ontology [18] describes terms used by the Bologna process. It can represent the departments, the teaching units together with information about their ECTS credits and teaching language. It can also be used to store students' examinations, their results and degrees. Although it aimed at providing a standard schema for European universities, in our modeling work we did not find any use of it.

The main goal of this study was to reveal the usability of the above ontologies in our use case. The following general consequences were drawn:

- It is crucial to understand the meaning of the main concepts and the relationships among them in case of each ontology. Unfortunately, in most cases these ontologies use essential concepts without defining their meaning (i.e. what do concepts like course, subject, module, study programme exactly mean and how do they relate with each other?).
- A specific ontology is usable only if the definitions of the main concepts fit into the use case. Furthermore, in the decision process it is important to see what kind of implementation is used in case of a certain property or relationship (e.g. defining temporal information of

a course can be achieved in various ways but does the actual one satisfy our requirements?).

- The correctness of the formal description of the ontology is important. If it contains shortcomings and mistakes, its reuse is cumbersome.

Currently, there is no other ontology suitable for the use case scenarios described earlier. Existing ontologies miss properties and thus cannot provide a full description of teaching activities. Furthermore, existing ontologies contradict each other in the naming and semantics of subject, course, curriculum, etc.

AIISO	**Teach**	**XCRI-CAP**	**MLO-Adv**	**Bowlogna**
Programme	Study Program	Qualification	Learning Opportunity Specification	Study Track
Subject	Course	Course	Learning Opportunity Specification	Subject
Course	Course	Presentation	Learning Opportunity Instance	Teaching Unit

Table 3: Core terms mapping between the different education-related ontologies

Basically, we had to find a set of ontologies filling all capability columns with a minimal number of overlaps. The selection criteria were also determined by several rational considerations, like availability, maintenance, usage and modularity. Before the final selection was made, we had to harmonize the term usage and adapt terms to the Hungarian system if it was possible at all. The final decision was to base our work on the AIISO ontology, because the structure of the concepts Course-Subject-Programme in this ontology fits into the Hungarian system the most.

Table 3 demonstrates how different ontologies use different labels for more or less similar concepts. For clarity, we give a short summary of how these terms are interpreted in Hungarian education.

After enrolling to the university each student is assigned to a **Curriculum** (in Hungarian “tanterv”), which is a set of **Subjects** (in Hungarian “tantárgy”) and their relations (i.e. dependencies among the subjects). Curriculum might specify **Specializations** (in Hungarian “specializáció”), which are sets of compulsory and optional subjects. Each Curriculum has a specific **Attendance Pattern** (in Hungarian “munkarend”, e.g. full-time, part-time, correspondence), and a result as a specific **Degree** (in Hungarian “fokozat”, e.g. BSc, MSc, BA, MA, PhD). A Curriculum is in many-to-one relationship with a **Study Programme** (in Hungarian “szak”, e.g. Computer Science Engineer), offered by the university. The curriculum is the specification how the Study Programme can be completed. A Study Programme determines the qualification that a student will get after the successful completion of his/her studies. A Study Programme must be accredited by an external body. A Curriculum is valid for a given time interval, meaning that a student can be assigned to it only if his/her enrollment time falls into this period. For each Subject, there is an **Organizational Unit** responsible for it. **Courses** (in Hungarian “kurzus”) are advertised based on a Subject, have temporal (**Course Time**) and spatial (**Location**) attributes and one or more assigned **Teacher**(s).

4.4 Methodology

The five-star model of good Linked Data vocabulary use [37] acted as a guideline during our work. As reaching to the 5th star it might take significant time, our initial aim was to design a 4-star vocabulary. The following rules were applied to restrict the potential interpretations of the defined classes and properties towards their intended meaning:

- Dereferenceable human readable information should exist about the ontology (e.g. a web page documenting it).
- The ontology should be described by a formal language, like OWL.
- The ontology should be linked to other ontologies.
- The ontology should contain metadata about itself (e.g. authors, modification date, used ontology language, status of the ontology terms, license information, etc.). All these ensures the adoption of the model.

The last condition, the fifth star was not relevant for us in the initial work, as we could not ensure that other vocabularies are linking to OLOUD, as these links have to be on the vocabulary level. In this case, the creator has limited influence as it reflects the external usage and perceived usefulness. However, since 4-star Linked Data vocabularies are already powerful and easy to use, and they will more likely earn the fifth star in the future.

OLOUD was developed based on the Uschold and King methodology, which consists of the following steps [73], [74]:

1. Identify the objectives of the ontology development and the intended usage (see Section 4.2); determine the necessary formalization level (see Section 4.6).
2. Specify the ontology by outlining the domain. This includes the identification and the clear textual definition of key concepts and relations (see Section 4.1). Furthermore, setting up identifiers for concepts and relations is necessary.
3. Formalize the terms defined in the specification using a formal language (see Section 4.6).
4. Integrate with existing ontologies. During specification and formalization, it is an important step to research third party ontologies for potential reuse and inclusion (see Section 4.5).

5. Evaluate the fruition of the objectives and the completeness of the ontology based on a predefined (generic and ontology specific) criteria (see Section 4.7).
6. Specify the documentation principles, which should be aligned to type and objective of the ontology (see Section 4.6).

4.5 Integration with other ontologies

In the process of creating a schema for a Linked Open Dataset it is advisable to reuse the available ontologies or vocabularies as much as possible. There are quite different vocabulary reuse strategies [57]. The two basic forms are (1) reusing classes and properties from existing vocabularies directly, and (2) establishing links at schema-level. The second case means defining new classes as either subclasses or equivalent classes and properties as sub properties or equivalent properties of the classes and properties of the reused ontology. The reuse strategies can be influenced by various factors, like reuse only one (or a few) domain specific vocabulary to provide a clear data structure or reuse only popular vocabularies to make the data easier to be consumed.

During development of the OLOUD ontology our strategy was the following: First the necessary concepts (classes and properties) were identified. Then reusable vocabularies, which could serve to express the defined concepts, were chosen according to criteria such as wide usage, OWL 2 compatibility, and regular maintenance. Based on our study of available vocabularies, which was detailed in Section 4.3, we decided to use AIISO as the base structure and then use other ontologies to fill in the gaps. Ontologies that are not specific to education are used to represent general concepts and relations, such as resource metadata (Dublin Core), people (FOAF), time (W3C Time and Temporal Aggregates Ontology [48]), events (Event [53]), address (vCARD [36]) and indoor location (iLOC [72]). In case of necessary classes and properties missing from the previous ontology list, new OLOUD terms were introduced. If it was possible, the new terms were linked on schema level to the above ontologies with the *rdfs:subPropertyOf* or *rdfs:subClassOf* properties.

Integration work posed the problem of fragmentation. In several cases an ontology was needed only for a single property (e.g. address). FOAF and vCard are similar ontologies, but each lacks some important properties, and thus both had to be used to fill in the holes. In the integration process, it was revealed that too many ontologies were needed to express desired goals, and some ontologies were hard to reuse because of inaccuracy or inconsistency in them.

4.6 Ontology description

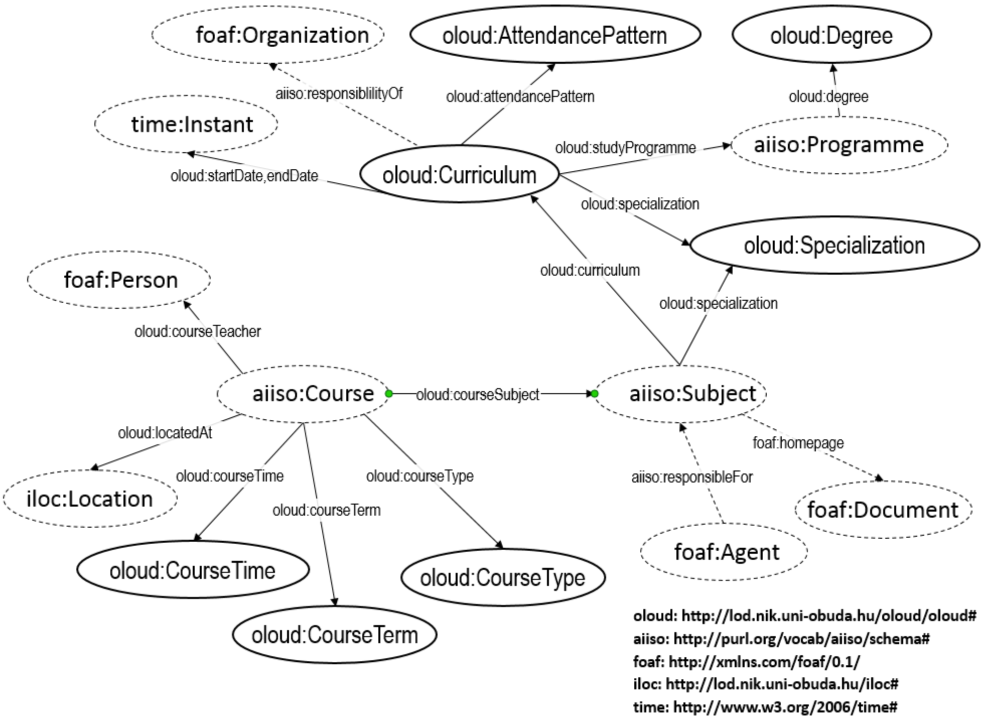


Figure 4: Overview of the main classes and properties in OLOUD

Figure 4 represents the overview of the new ontology: the main classes, the highlighted object properties connecting them and the essence of the class hierarchy as well.

Classes defined in OLOUD are using the *oloud* prefix, classes and properties needed from other ontologies are using their own prefix and marked with a dashed line on the diagram. The most important classes in OLOUD are *Curriculum, Subject, Course,* and *Programme*. The Curriculum class is defined as a subclass of the *aiiso:KnowledgeGrouping* class, Subject, Course and Programme classes are used directly from AIISO. In the following, these classes are described with their direct connections.

Curricula at Hungarian universities contain the list of subjects with their dependencies (i.e. each subject can have various prerequisite subjects). A specific Curriculum entity is connected to the Subject entities with the *curriculum* property. The faculty or department of the university responsible for the Curriculum is linked with the *aiiso:responsibilityOf* property. Each Curriculum has a period of validity determined by the *startDate* and *endDate* properties. The possible Specializations in each Curriculum are determined with the *specialization* property. The Attendance pattern, the Study programme and the Degree of the study are set by the *attendancePattern, studyProgramme* and *degree* properties. The language of the studies according to the Curriculum is given with the *dcterms:language* property.

Subjects are featured by their name, code, number of credits, person, and responsible organization: *foaf:name, aiiso:code, subjectCredit, aiiso:responsibilityOf.* A Subject entity belongs to a specific Curriculum entity. The connection between a Subject and its Courses is set by the *courseSubject* property defined in OLOUD, since AIISO does not provide any property to connect these concepts. The prerequisite conditions between Subject entities are set by the *subjectRequires* property.

The entities of the Course class are the actual instances of subjects having spatial, temporal and type descriptions, identification number, name, and instructor: *locatedAt, courseTime, courseTerm, courseType, aiiso:code, foaf:name,* and *courseTeacher*. To properly describe entities of the Curriculum, Subject, and Course classes, some auxiliary classes were introduced: *StudyProgramme, Degree, AttendancePattern, Specialization, CourseTerm, CourseType*. The aiiso:Programme class is used to represent the Study Programme concept, Specialization is defined as a subclass of the aiiso:Module.

To enable step-by-step indoor navigation, the *Location* class from the iLOC ontology is being used to represent the necessary entities describing indoor locations for Course and Event entities. Courses and events can be assigned to Rooms, and Rooms are connected via a network of POIs (Points of Interest), which can be doors, hallway connections, etc. The offices of lecturers and other significant and popular rooms can also be included in the description of campus buildings.

Entities providing a temporal description of courses in OLOUD are based on OWL-Time and Temporal Aggregates Ontologies. Our objective was to enable SPARQL queries according to date, time and duration and to define course time as recurring events. These objectives can be satisfied with the above ontologies. We suggest using a separate ontology module for the 'time related' concepts. In this module subclasses are defined for classes in OWL-Time and Temporal Aggregates Ontologies, facilitating the generation of entities describing recurring events.

The OLOUD ontology consists of two modules: OLOUD-BASE [69] and OLOUD-TIME [70]. The former describes all the 'university related' concepts, uses the prefix *oloud* and namespace *http://lod.nik.uni-obuda.hu/oloud/oloud#*. The latter provides the necessary classes and properties to describe course time data as recurring events, uses the prefix *otime* and namespace *http://lod.nik.uni-obuda.hu/oloud/otime#.* In OLOUD-BASE there are 7 classes, 16 object properties, 5 data properties, and 14 individuals, while in OLOUD-TIME 6 classes are defined at this moment.

OWL 2 RL was chosen as the formal language for OLOUD. The advantage of this ontology approach is that new classifications can be inferred by rules and class restrictions, such as subjects announced for the current semester, subjects meeting the prerequisite criteria in case of a specific student or course announcements having various properties. The OLOUD Ontology was implemented in a self-documenting way. Based on the request MIME type it can be downloaded in different formats including the human consumable HTML output, which is automatically generated from the following properties: *rdfs:label, rdfs:comment, rdfs:domain, rdfs:range*. The ontology description was implemented as metadata best practice described in [75], by adding the recommended metadata instances and addressing the outlined policies. The Ontology is licensed under the terms of Creative Commons 3.0

4.7 Example data

Based on OLOUD, a linked dataset was built for Óbuda University. The location data was created manually based on building layout diagrams of the university, while the subject and course data were converted using custom scripts from relational database dumps extracted from the electronic administration system (Neptun) of the Óbuda University. The entities of the Person class and their personal data were scraped from personal webpages. The university event descriptions were generated with a crawler from the OU webpage. The dataset was also extended with links to the GeoNames geographic dataset. As an example, we provide a sample Course and Subject description:

```
:AB0_LA_01_E_2014-15-1 rdf:type aiiso:Course,
aiiso:code "AB0_LA_01_E"@hu ;
foaf:name "Adatbázisok"@hu ;
oloud:courseSubject :NAIAB0SAED ;
oloud:locatedAt :PC_Labor_220 ;
oloud:courseTeacher :Dominika_Fleiner ;
oloud:courseTime
odata:CourseTime;courseTerm=2014Fall;hour=17;minute=55;durationHour=
1;durationMinute=35;dayofweek=1 ;
oloud:courseType :Lab ;
oloud:courseTerm :2014Fall .

:NAIAB0SAED rdf:type aiiso:Subject,
oloud:subjectCredit "4.00"^^xsd:Integer;
foaf:homepage "http://users.nik.uni-obuda.hu/to/tantargy/
adatbazisok-0" ;
aiiso:code "NAIAB0SAED" ;
foaf:name "Adatbázisok"@hu ;
aiiso:responsibilityOf :AII ;
oloud:studyProgramme :ComputerEngineer ;
aiiso:responsibilityOf :Dominika_Fleiner ;
oloud:attendancePattern :FullTime ;
oloud:specialization :ObligatoryPart ;
oloud:degree :bsc;
oloud:curriculum :MérnökInformatikusBSc_2012 .
```

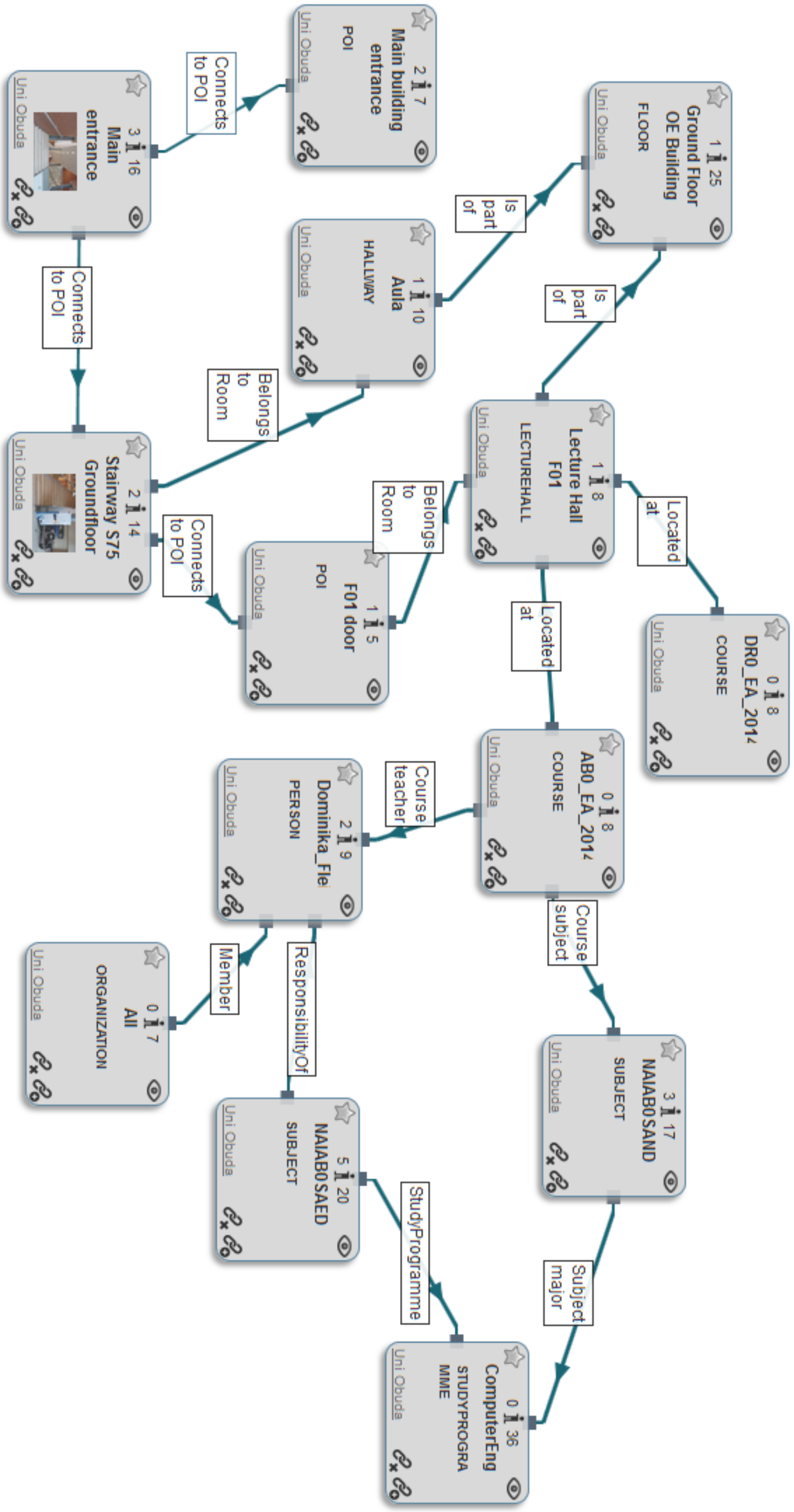


Figure 5: Extract of the dataset in the Lodmilla browser

Figure 5 provides a visualization of a part of the linked data graph. Browsing the dataset with a LOD tool such as LODmilla [44] provides access to useful data for students. On the figure, there are two different subjects belonging to the same Study Programme, one of the subjects has a course linked, which is taught by the same person responsible for the other subject. In the graph, there are a few locations linked together: two courses are scheduled in the F01 room, which is on the ground floor of the Óbuda University building. The room has a door and connects to the Aula, which has an entrance and a connection to the stairway.

Furthermore, an in-door wayfinding service for students is also available, providing a simple itinerary to the place of the given lecture. It is available as a mobile app backed by an open SPARQL endpoint.

Using this dataset, we can express in SPARQL a wide range of practical queries for combinations of course information, time and location data. It is possible for example query the occupation of a lecture room on a day, get all lectures for a given course in this semester, or all lectures for a student today based on her course list.

Additionally, it can be decided if there are overlapping courses in the semester, with a query like the following:

```
SELECT ?course1 ?course2 {
?course1 oloud:courseTime ?ct1.
?ct1 ta:hasTemporalAggregateDescription ?ta1.
?ta1 ta:hasithTemporalUnit ?day;
ta:hasStart ?s1.
?s1 time:hasDateTimeDescription ?dtd1 .
?dtd1 time:minute ?minute1 ;
  time:hour ?hour1 ;
  time:hasDurationDescription ?duration1.
?duration1 time:hours ?durationHour1;
 time:minutes ?durationMinute1.
```

```
?course2 oloud:courseTime ?ct2.
?ct2 ta:hasTemporalAggregateDescription ?ta2.
?ta2 ta:hasithTemporalUnit ?day;
ta:hasStart ?s2.
?s2 time:hasDateTimeDescription ?dtd2 .
?dtd2 time:minute ?minute2 ;
  time:hour ?hour2 .
BIND (xsd:integer(?minute1)+(xsd:integer(?hour1)*60) AS ?start1).
BIND (xsd:integer(?minute2)+(xsd:integer(?hour2)*60) AS ?start2).
BIND
(xsd:integer(?durationMinute1)+(xsd:integer(?durationHour1)*60)+
?start1 AS ?end1).
FILTER (?start1 < ?start2 && ?start2 < ?end1).
FILTER (?course1 != ?course2 ).
}
```

4.8 Conclusions

The original idea was to implement useful, “smart” services for university students based on linked data. We realized that there are many ontologies or vocabularies for the domain, yet none of them is suitable for our complex purpose. The biggest problems we found were the missing distinctions between subjects and courses and between curricula and study programs. We created the OLOUD ontology which extends and amalgamates selected ontologies and facilitates the full description of course-related information. Hence, we think our work helps to better clarify the role of frequently used concepts in this domain.

5 The iLOC Ontology

This chapter presents the iLOC ontology as a generic approach for indoor route finding and location description. The main motivation for developing a new ontology was born during the implementation of a university campus map and indoor plan for buildings as linked data. The conclusion of a survey of the existing domain ontologies was, that there is no available ontology for this purpose even with limited capabilities. This recognition gave the purpose of the new ontology development.

The suggested Linked Data model enables the integration of various datasets related to indoor spaces and can be used in museums, shopping malls, hospitals, campuses, stadiums, airports, etc.

iLOC supports three levels of details for a building plan, with increasing complexity and capabilities. Furthermore, routes can be calculated with flexible constraints for users' goals and abilities. The ontology is demonstrated with a smartphone application helping university students to find their lectures. Example SPARQL queries for route calculation were provided and the performance has been measured with different SPARQL engine implementations.

5.1 Modeling Indoor Spaces with Ontologies

Worboys [76] provides a thorough overview of the state of the art and defines a top-level taxonomy to classify indoor models into semantic and spatial categories. Semantic indoor space models represent entity types, their properties, and relationships. Topological models are concerned with the connectivity within a space. Geometrical models add quantification of distance and finally, hybrid or multilayered models provide combined features of all the above.

Although OGC (Open Geospatial Consortium) provides a standard open model called IndoorGML [47] built on top of CityGML [39], these are XML based, and therefore their use as linked data is not feasible. The INSPIRE[20] directive aims to create a spatial data infrastructure in the European Union. It has a very complex data model covering transport networks and buildings as well but the inside of buildings is not covered yet. Furthermore, there is no accepted way of using INSPIRE as linked data [43].

In the following, we review the field of indoor navigation ontologies. OntoNav [3] is a semantic indoor navigation system and an ontological framework of handling routing requests. OntoNav navigates the users inside floors and buildings, but it does not provide navigation instructions within rooms. In the case of a large hall with many entrances, it is useful to have routes inside the hall as well. iLOC navigates on a POI network, so it offers a more generic solution.

ONALIN [20] provides routing for individuals with various needs and preferences; it takes the ADA (American Disability Act) standards, among other requirements, into consideration. Buildings are modeled as hallway networks, and feasible routes can be identified for users having specific constraints.

The above implementations rely on custom reasoning algorithms and do not provide navigation through SPARQL queries, as the primary purpose was not linked data support. None of the above ontologies is accessible at the moment of writing this thesis, however, some parts of the conceptual semantic model

---

[20] http://inspire.ec.europa.eu

were reused from these earlier works inspiring the hierarchy of the *iloc:Location* class.

The classification of different parts of a building is out-of-scope for our ontology. One can find multiple linked data sources for such purpose (e.g. DBpedia has a list of room categories). The best source would be the OmniClass Construction Classification System (OCCS)[21], which is a classification system for the construction industry. Unfortunately, there is no OWL/RDF representation for OmniClass.

There are efforts, for example [42], where graph algorithms such as Dijkstra shortest path algorithm is run on semantic location data, which requires exact coordinates for all indoor positions. iLOC provides a simpler and less labor-intensive solution for route finding.

None of the above ontologies is accessible at the moment of writing this paper, which is a requirement for Linked Data publishing. However, some parts of the conceptual semantic model were reused from these earlier works inspiring the hierarchy of the *iloc:Location* class.

5.2 Ontology specification

The objective of our ontology is to allow publishing indoor location and navigation data as Linked Open Datasets and to support the development of applications using the published data. The ontology facilitates location description and links to relevant other data. It also offers concepts to describe how two places are connected to each other, including their distance or constraints on the route (e.g. a route section is not recommended for wheelchair users). Our requirements also included the low implementation cost, as well as supporting a broad usage (the solution should not require any special device to

---

[21] http://www.omniclass.org

consume and leverage the data, it should be able to operate with a generic smartphone or tablet).

According to the classification of Worboys, iLOC is a hybrid ontology that supports entity information, connectivity and distance as well. iLOC provides cost-effective building modeling and navigation based on simple SPARQL queries.

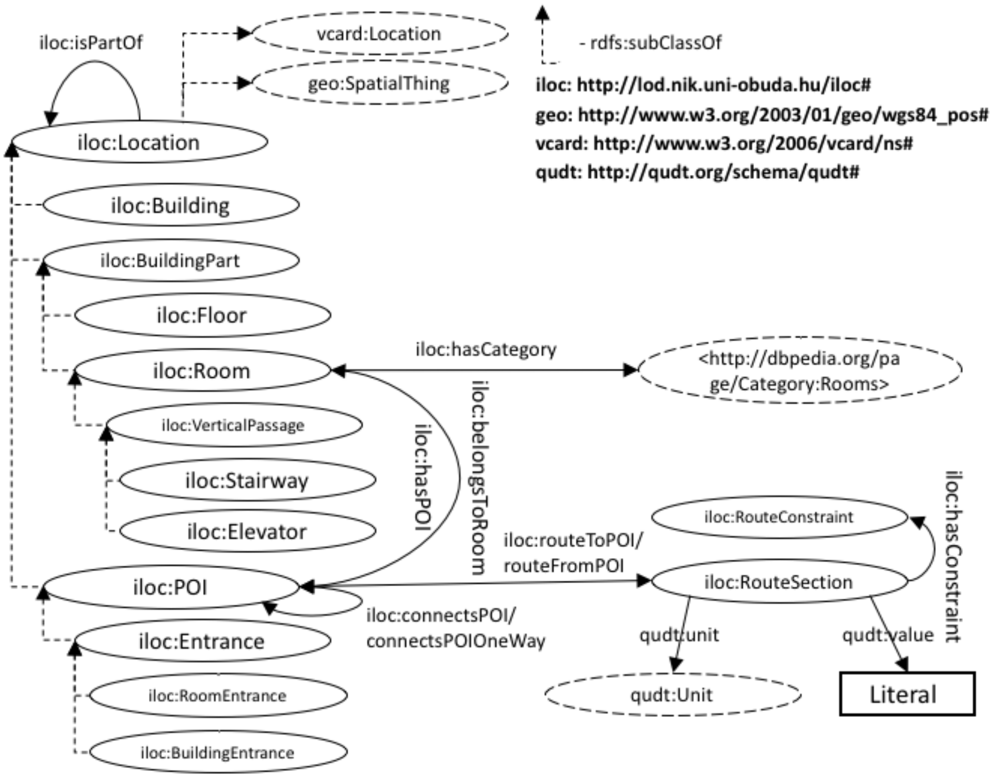


Figure 6: The overview of the iLOC ontology

Figure 6 shows the main classes (as ovals) and the most significant object properties (as arrows) between the classes. Classes defined in this ontology are using the *iloc* prefix, classes and properties used from other ontologies are prefixed with their own and such classes are marked with a dashed line. The defined classes can be used to describe buildings with the significant parts of the internal structure: rooms, floors, vertical passages and Point of Interests

(POIs). The POIs have a special role in the navigation, as these are the elements of an indoor route. Among OWL features, the most necessary in our case is the definition of symmetric properties (connectsPOI), although it can be managed using more complex SPARQL queries. We found that there are several transitive properties in iLOC (e.g. isPartOf, connectsPOI), but transitivity can be gained by SPARQL property chains. There are options for using rules to complete missing connections, for example, if two POIs are connected via a RouteSection, then connectsPOI can be inferred between the two POIs. Such rules can minimize the number of triples which need to be explicitly defined in a building model.

The *Location* class represents all the necessary entities describing indoor locations. It has three direct subclasses: Building, Building Part and POI. Entities of the *Building* class have an internal structure that the ontology aims to describe. *BuildingPart* provides an abstract concept to the different parts of the internal structure of a building. It has two subclasses: *Floor* and *Room*. Entities of the Room class can be classified into further external categories by the *hasCategory* property or by defining specific subclasses. A category can be anything; it is not limited to be an RDF Class. The room categories are not used in navigation; these can be used as custom filters to get a list of specific type of rooms. Typical other Room categories would be Lecture Hall, Office, Canteen, Toilet, etc. The Room class has a subclass: *VerticalPassage*, which has two further subclasses: *Stairway* and *Elevator*. They connect Floors, which can be exploited in the simplest form of our navigation methods.

The *POI* (Point of Interest) class has a significant role in the navigation, as POI entities constitute the elements of a classical indoor route. The *POI* class has one subclass: *Entrance*, which is further specified as *RoomEntrance* and *BuildingEntrance*. The Entrance instances have the special meaning of defining the connections between rooms or the entry point to buildings.

Similarly, to the class *Room*, entities of the *POI* class can be classified into further external categories by the *hasCategory* property. In our university use case, POIs would also contain 'landmarks' inside the buildings such as statues, vending machines, receptions, and favorite meeting points.

The *Route Section* class represents a traversable path between two POIs, the *Route Constraint* class represents any permission or ability required to traverse a Route Section. Such Route Constraints can be a required employee badge or the ability to climb stairs.

The *isPartOf* object property expresses the hierarchical structural relationship in the building, e.g. a specific Room entity can be in isPartOf relation with a specific Floor entity. The *connectsPOI* property describes a direct route between two POI instances, and a consecutive sequence of such direct routes generate possible navigation steps between two POIs. The *connectsPOIOneWay* is the asymmetric parent property of the connectsPOI property, to describe one-way routes between points.

The *hasPOI* property and its inverse property (*belongsToRoom)* express POI and Room relationships, the specific Room entity contains the given POI entity. The *hasCategory* property can be used to describe rooms or POIs by (external) categories, stating for example, that a Room entity is a Cafeteria. Finally, the *hasAccess* property can be used to associate a specific constraint to a RouteSection instance, requiring permission or ability for one to traverse it. The *distance*, *stepNumber* and *incline* properties can add extra information to a specific Route Section that can be used in patient customized wayfinding queries. Metrics for the distance property value can be specified by a unit with the QUDT ontology (providing generic measures and units to reuse). The walking distance and walking time would be the natural metrics here.

The *defaultRoomOf* property provides a location description for *foaf:Agent*s (Persons and Organizations). With the help of this property, personal offices or reception rooms can be specified.

The Ontology is licensed under the terms of Creative Commons 3.0[22]

5.3 Integration with other ontologies

By integrating iLOC with other ontologies, navigation queries can be enriched with filtering closures defined by concepts from third-party ontologies. These can define room categories or accessibility options and route types.

The root level class (*iloc:Location*) is a subclass of the *geo:SpatialThing* class reusing the W3C Basic Geo Vocabulary[23] in order to describe the real-world location of our entities in a standard way. As it was explained in the Related Work section, we had no opportunity to reuse any existing indoor location ontologies.

The vCard Ontology[24] describes the addresses of buildings. The QUDT ontology[25] provides properties for quantifying the route sections (distance, walking time, etc.). The QUDT ontology also defines a large set of metric unit definitions for reuse.

The LinkedGeoData ontology provides a long list of POI subclasses; although most of them only apply to outdoor use cases, few of them are applicable indoor too: VendingMachine, WasteBin, etc. If there is a need for a custom POI type, one can create corresponding subclasses of the iloc:POI class.

---

[22] http://creativecommons.org/licenses/by/3.0/

[23] http://www.w3.org/2003/01/geo/#vocabulary

[24] http://www.w3.org/TR/vcard-rdf/

[25] http://qudt.org/schema/qudt#

5.4 Extension points

iLOC provides multiple options for extension:

- Each building is a *geo:SpatialThing*, so indoor and outdoor spaces can be linked.
- The general-purpose Room and POI classes can be further specified by subclasses or using the *iloc:hasCategory* property, which can point to room and POI categories defined in DBpedia or to other custom categories.
- Similar extension point is the *iloc:AccessFeature* class, which refers to any type of restricted areas, requiring permission to enter, or specifies abilities required to traverse a route section. Specific constraints can be defined as instances of this class. iLOC defines some basic example instances, which can be used to model accessibility for wheelchair users.
- Metrics for a RouteSection instance can be specified by a unit and a value. The QUDT ontology provides generic measures and units to reuse. The walking distance and walking time would be the natural metrics to use here, but other extensions are also realizable.
- Further description of buildings and building parts is also possible. Photos can be linked using the *foaf:depiction* property. Links using *rdfs:seeAlso* can point to web pages or other linked data entities (such as DBpedia pages)

It can be useful to add exact geolocation data to POIs using a geospatial ontology such as the W3C Geospatial Vocabulary[26] which supports not only coordinates but floor number and altitude as well.

---

[26] http://www.w3.org/2005/Incubator/geo/XGR-geo/

5.5 Conclusion

The iLOC ontology enables data providers to publish navigation data about buildings. It supports three levels of capabilities, which are building on each other and provide more features but also require bigger investment during the modeling phase. The ontology provides the advantage of having a single RDF data model and thus pathfinding can be combined with additional semantic queries.

Our goal was to enable a wide variety of location-based information tools with the help of linked data. iLOC can provide very simple and easily maintained datasets for navigation, and the use of third-party geospatial databases and algorithms can be avoided.

6 Indoor Navigation with distributed Linked Data

Outdoor navigation is widely available nowadays and helps people to find a place while driving or walking or using public transport. This type of navigation is usually based on coordinates provided by GPS. Inside buildings, however, a navigation system has to cope with more complex routes and with the lack of GPS signals. The increased growth of location-based services and the fact that people spend their time mostly indoors lead to strong demand for indoor space applications. The current solutions usually require special and expensive hardware for indoor positioning. Thus, opposed to the outdoor scenario, there is no single, accepted basic methodology for navigation. The typical use cases include supporting students at a university in finding their next lecture, guiding patients to their destination within a hospital, supporting shopping in a mall or navigating to a gate at an airport.

There are many algorithms for outdoor path routing, but indoor routing algorithms are not so diverse. Dijkstra's shortest path algorithm is being used at most of the researches.

6.1 Graph route-finding tools

Reachability and shortest path problems have been intensively studied in the graph community. A survey of this research can be found in [2]. There are efforts, for example [3], where graph algorithms such as Dijkstra shortest path algorithm is run on semantic location data. Several widely used tools can be applied to this problem, this section elaborates on route finding with SPARQL 1.1, with the Virtuoso transitive SPARQL extension, with Gremlin.

6.1.1 SPARQL

SPARQL 1.1 supports property path queries. It does not return what the path is nor the length of the shortest path - only whether there is such a path. By probing against different path lengths, this limitation can be overcome by a query similar to the following, which returns the shortest route (routes with the least steps) between `<room1>` and `<room2>` (with a maximum length of three steps for brevity):

```
PREFIX iloc: <http://lod.nik.uni-obuda.hu/iloc/iloc#>
SELECT ?distance ?start ?p1 ?p2 ?p3 ?end WHERE {
  BIND (<room1> AS ?start ).
  BIND (<room2> AS ?end).
  ?p1 iloc:belongsToRoom ?start.
  ?p1 iloc:connectsPOI ?p2.
  ?p2 iloc:connectsPOI ?p3.
  ?plast iloc:belongsToRoom ?end.
  FILTER (?p3 = ?plast || ?p2 = ?plast || ?p1 = ?plast )
  BIND (if( ?p3 = ?plast , 3, if( ?p2 = ?plast , 2, if( ?p1 =
?plast , 1, -1))) AS ?distance)
}
ORDER BY ?distance
LIMIT 1
```

Future SPARQL versions might better support such path queries by enabling access to the length of the path or to the specific elements of a route.

6.1.2 OpenLink Virtuoso transitive extension

Jena introduces a path extension for SPARQL. This allows, for example, saying that `{<Alice> foaf:knows* ?x}`, meaning that `?x` is bound to the transitive closure of all people `<Alice>` knows. OpenLink Virtuoso takes a more general approach and allows an arbitrary subquery to be made transitive.

This has the advantage of being able to also retrieve properties of steps and to have complex conditions for what conditions define relatedness.

The transitive closure operator can only compute transitivity when starting in a given URI. This is not a significant limitation for the navigation use cases, as usually the starting point of a route is known. The following SPARQL query provides the same result as the previous example:

```
SELECT ?path ?distance ?link WHERE {
{
  BIND (<room1> AS ?start ).
  BIND (<room2> AS ?end).
  SELECT * WHERE { ?start iloc:connectsPOI ?end. }
} OPTION (TRANSITIVE, t_no_cycles, t_shortest_only,
  t_in(?start), t_out(?end), t_step (?start) as ?link,
  t_step('path_id') as ?path,
  t_step('step_no') as ?distance, t_direction 3  ).
}
```

6.1.3 Gremlin

Gremlin [55] is a graph traversal machine and language designed, developed, and distributed by the Apache TinkerPop project. Gremlin, as a graph traversal machine, is composed of three interacting components: a graph, a traversal, and a set of traversers. The traversers move about the graph according to the instructions specified in the traversal, where the result of the computation is the ultimate locations of all halted traversers. A Gremlin machine can be executed over any supporting graph computing system such as an OLTP graph database and/or an OLAP graph processor. Gremlin, as a graph traversal language, is a functional language implemented in the user's native programming language and is used to define the traversal of a Gremlin machine.

The purpose of the language is to enable a human user to easily define a traversal Ψ, which is a tree of functions called steps, and thus, program a Gremlin machine. The following Gremlin code fragment provides the same result as the previous examples:

```
start = g.v(<room1>)
end = g.v(<room2>)
start.as('x').dedup().out('iloc:connectsPOI').loop('x')
  { it.loops < 3 && !it.path.contains(it.object) &&
    it.object != end }
.path.filter{it.last()==end}[0]
```

6.2 The iLOC ontology in use

The iLOC ontology enables the integration of various datasets related to buildings and can be used in museums, shopping malls, hospitals, campuses, stadiums, airports, etc. Possible use cases include the description of nearby places, planning routes or finding places with given features. iLOC supports three levels of details for a building plan, with increasing complexity and capabilities: one based on the building hierarchy and two on POI networks within the building. Furthermore, routes can be calculated with flexible

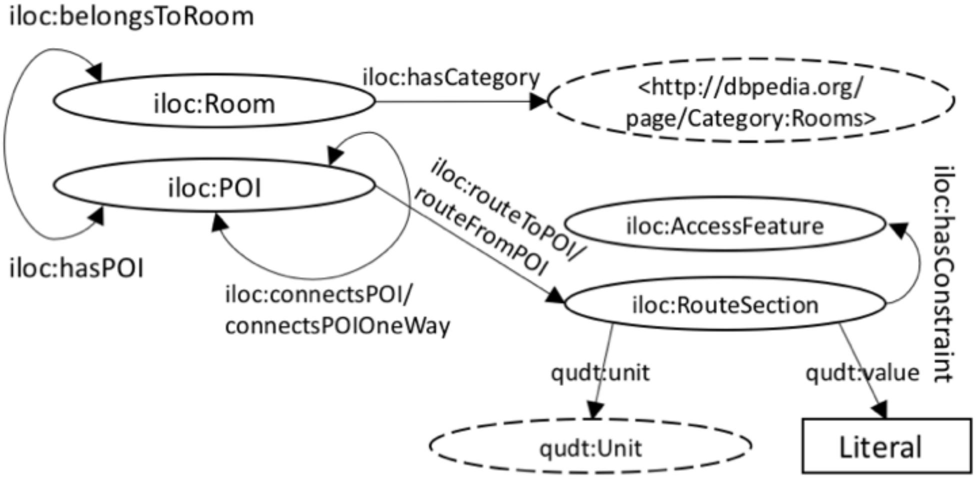


Figure 7: The topological model

constraints for users' goals and abilities. iLOC is a hybrid ontology supporting both taxonomical and topological description of indoor spaces.

Figure 7 illustrates the topological model, which supports the path based indoor navigation within the POI network with two different level of detail. In the first approach, a path consists of several iloc:POI instances linked together with *iloc:connectsPOI* (or *iloc:connectsPOIOneWay* – in case of one-way pathways) properties. The actual room can be inferred using the *iloc:hasPOI* properties. The other way of modeling a traversable path is using route sections, which provide constructs for a more accurate description: one can describe the accessibility of the path (e.g. wheelchair accessible) and also provide some metrics about the path (e.g. length in different measurement units or the approximate time it takes to walk through).

6.3 Navigation

iLOC supports three navigation methods, with different complexity and accuracy. While the first method only informs the user which building, part of building and floor they should look for, the second is able to provide the actual steps it takes to navigate through the building and the third method provides the most accuracy by describing the dimensions of the steps as well as some of the restrictions one can face while being underway.

6.3.1 Simple navigation based on the building structure

The first method is based on the structure of the building (and based on the *iloc:isPartOf* property) and provides simple instructions like *„Enter the main building. Go to the 3rd floor. Look for Office 307.”*

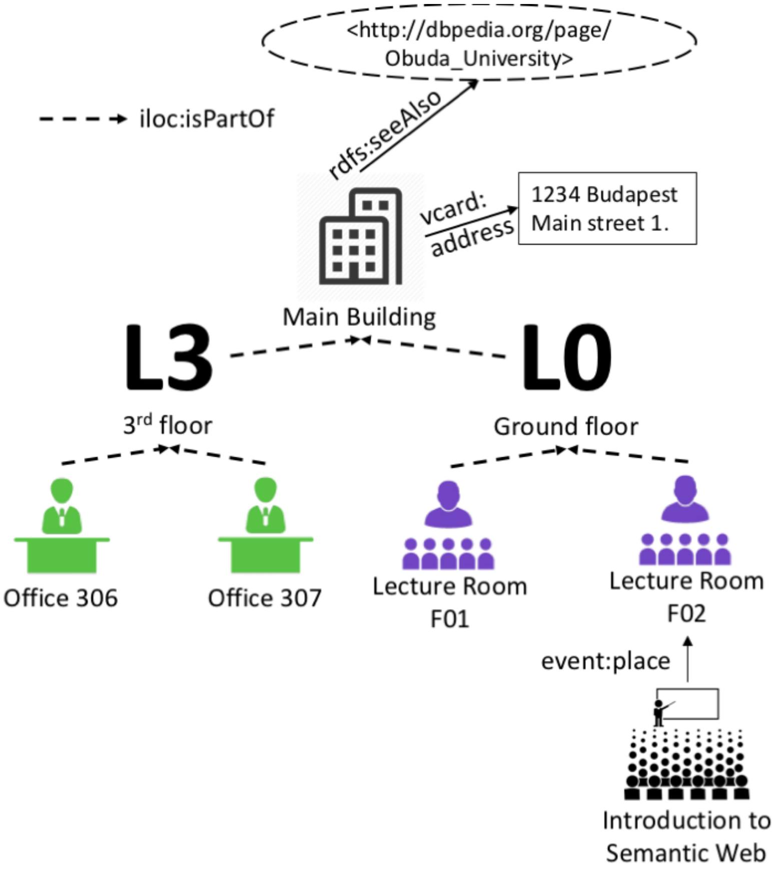


Figure 8: Modeling building structure in iLOC

Figure 8 illustrates part of a building as an example. The building has an address and a DBpedia page associated. On the 3rd floor there are two office rooms. F01 and F02 are two lecture rooms on the ground floor. F02 hosts the Introduction to Semantic Web course, which is defined in a third-party dataset and linked to F02 through the Event Ontology[27].

6.3.2 Navigation based on POI networks

The second option builds on the top of the structural description, introducing special points in the space: „Point of Interest (POI)". One POI instance is

---

[27] http://motools.sourceforge.net/event/event.html

connected to one or multiple other POI by the *iloc:connectsPOI* property. The route is a chain of POIs within the POI network. As Rooms can be the target of a navigation task, it is important to relate them to the POI network. The *iloc:belongsToRoom* and *iloc:hasPOI* properties are responsible to connect the rooms to the POI network by the nearby or including POIs. The second approach is more aligned to the instructions one would expect asking someone familiar with the building: *„Enter the main hall. Pass the coffee machine. Go to the elevator. Go to the 2nd floor. Pass the restrooms. Look for Room 212."* While the second approach allows more accurate navigation, there is the extra cost of modeling the POI network compared to the first approach.

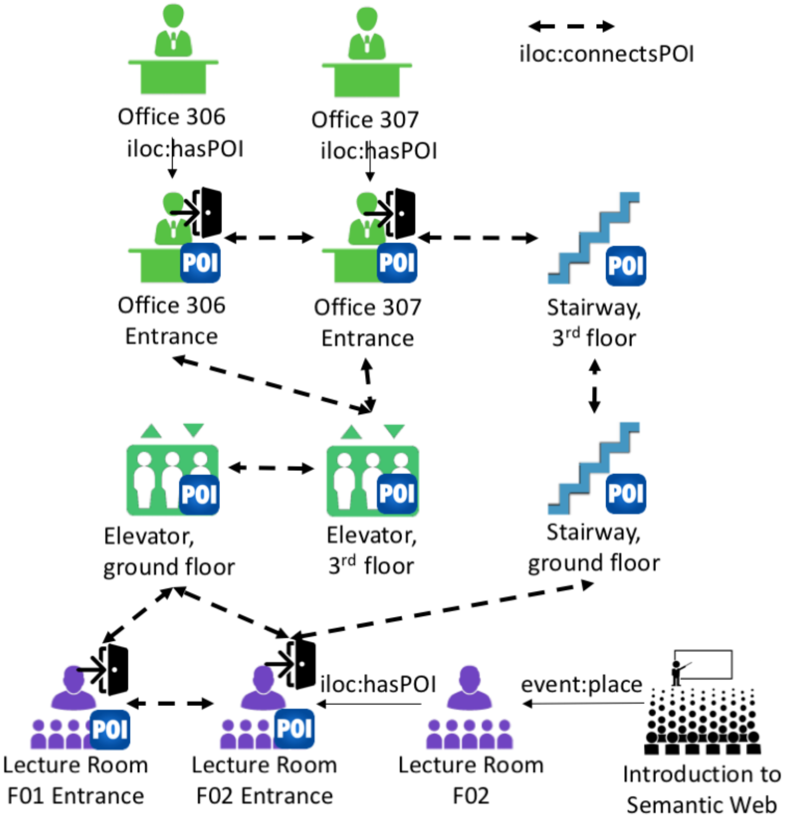


Figure 9: iLOC POI network example

Figure 9 illustrates a potential portion of a simple POI network of a building. In this example, the building has a floor defined with two offices and two lecture rooms on the ground floor. Each office and lecture room have one or multiple entrances, represented by POIs. Each POI is connected by the *iloc:connectsPOI* property, if there is a direct walkable route between them. One of the lecture rooms hosts the Introduction to Semantic Web course. An elevator and a stairway connect the different floors.

To navigate with linked data, SPARQL 1.1 is suitable, although the inference of symmetric properties is useful here.

6.3.3 Advanced navigation based on POI networks

The iLOC Ontology supports more advanced and accurate navigation, providing tools to describe the route sections as restricted, one-way or with metrics to quantify the length or time duration between the start and destination points. These are enabled by the *RouteSection* class: a *RouteSection* instance is associated with two POIs by the *iloc:routeToPOI* and *iloc:routeFromPOI* properties. A *RouteSection* might be linked to *RouteConstraint* instances (for

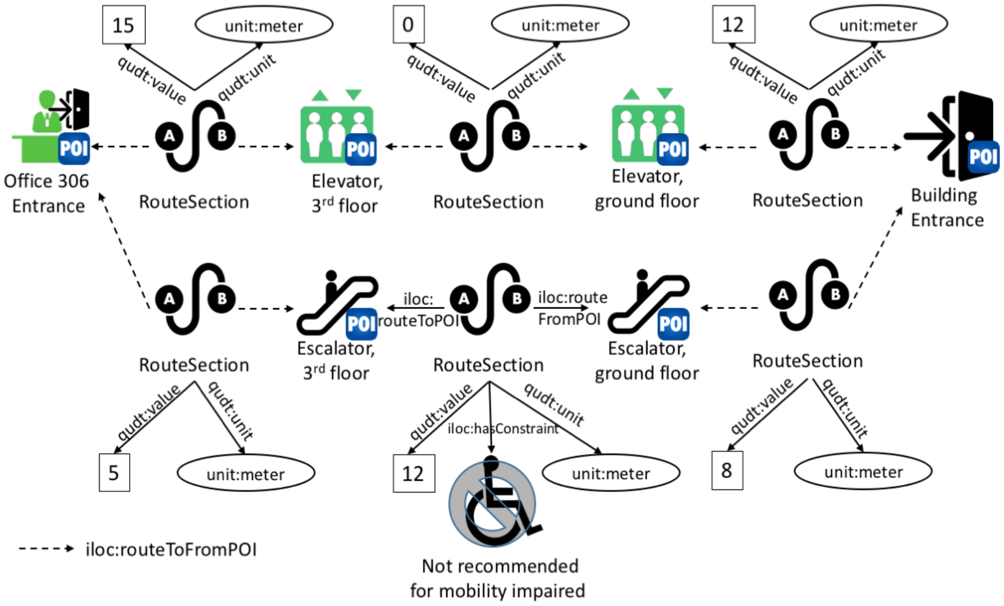


Figure 10: Modeling indoor routes with the RouteSection and RouteConstraint classes

example having a badge or physical ability). The QUDT ontology adds metrics to the route sections by the *qudt:unit* and *qudt:value* properties. The QUDT ontology also defines a set of units to reuse.

Figure 10 illustrates a complex example of using the *RouteSection* and *RouteConstraint* classes. In this example, there are two different routes defined between Office 306 on the third floor and the building entrance on the ground floor. The first route uses the elevator; the second one uses a one-way (ascending) escalator (for the sake of the example). The route is represented by POI instances connected by *RouteSection* instances. The property connecting these instances can be either *iloc:routeToFromPOI* representing a bi-directional connection or the pair of *iloc:routeToPOI* and *iloc:routeFromPOI* representing a one-way connection. On the example, the escalator is modeled as a one-way route from the ground floor up to the 3rd floor. Each *RouteSection* instance has a *qudt:value* and *qudt:unit* pair associated, which quantifies the length of the section. Using the *iloc:hasConstraint* property, the *RouteSection* representing the escalator is marked as "Not recommended for mobility impaired". The instance representing the constraint should be defined in a 3rd party ontology.

6.4 Finding and navigating to a specific nearby POI

Important use case to support when the user has no specific target in their mind or even the exact identifier or name of the location is unknown. One option is to query for every potential target location matching some specific criteria. As this query is domain dependent, it should be supported by 3rd party domain ontologies, and the result should be iloc:Room or iloc:POI URIs, which will be used in queries described in the previous sections. The other option is to combine the domain query and the navigation query to find the nearest location. The following example demonstrates how one can get from the main entrance to the nearest wheelchair accessible toilet.

```
prefix iloc: <http://lod.nik.uni-obuda.hu/iloc/iloc#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix ex: <http://example.org/>
SELECT ?distance ?sl ?l1 ?l2 ?l3 ?el WHERE {
  BIND (ex:MainEntrance AS ?start ).
  ?end iloc:hasCategory ex:Toilet;
      iloc:hasAccess iloc:WheelChairAccess.
  OPTIONAL {?start rdfs:label ?sl.}
  OPTIONAL {?p1 rdfs:label ?l1.}
  OPTIONAL {?p2 rdfs:label ?l2.}
  OPTIONAL {?p3 rdfs:label ?l3.}
  OPTIONAL {?end rdfs:label ?el.}
  ?start iloc:connectsPOI ?p1.
  ?p1 iloc:connectsPOI ?p2.
  ?p2 iloc:connectsPOI ?p3.
  ?plast iloc:belongsToRoom ?end.
  FILTER (?p3 = ?plast || ?p2 = ?plast || ?p1 = ?plast )
  BIND ((if( ?p3 = ?plast , 3, if( ?p2 = ?plast , 2, if( ?p1 =
?plast , 1, -1)))) AS ?distance)
} ORDER BY ?distance LIMIT 1
```

6.5 Query Performance

Although SPARQL is able to query longer path, not all implementation is capable to deliver results. This section summarizes our experiments with the Apache Marmotta and the Virtuoso triple stores and the Gremlin traversing engine.

The LOD server hosting the data of Óbuda University runs Apache Marmotta[28] and we also experimented with OpenLink Virtuoso, as its custom extension for property transitivity fits the task very well. In standard SPARQL 1.1, one cannot get the length of the property path, which would be very useful in this case. Instead, a complex FILTER clause can provide the length of the route. In some cases, Marmotta produced long response times in finding routes.

Appendix A summarizes SPARQL query efficiencies finding different routes in the graph.

Virtuoso was significantly faster with its non-standard extension. The best performance was measured with the Gremlin traversing engine, queries were returning results in the 150-250ms range.

The major challenge with the SPARQL 1.1 queries is to predict the length of the shortest route: we have observed the best performance when the query was looking for routes having the same length as the result. Having a standard SPARQL feature to determine property path length would be quite useful here to optimize query performance.

6.6 Conclusion

In this chapter, the potential role of Linked Open Data in indoor navigation was reviewed. Using an ontology, one can publish indoor maps in multiple layers of complexity. The iLOC ontology, introduced in the previous chapter, provides three layers. A more complex map is costlier to create but provides more accurate navigation. If such a map is linked to other LOD datasets, one can query for getting services from a not known location or get a more personalized route based on personal needs.

---

[28] http://marmotta.apache.org

7 Indoor modeling best practices with iLOC

When we ask someone the way to some places, people usually give route descriptions in an abstract and symbolic manner referring to different easy to identify landmarks or signs along the route. In order to provide a good indoor navigation experience, the guidance needs to integrate human wayfinding principles in route instructions. In this chapter, some of the best practices are summarized from the process of modeling one of the Óbuda University buildings.

7.1 Indoor modeling approaches

OntoNav [3] and ONALIN [20] employs a simple hallway network that represents the layout of each floor. In this network, hallways are represented by edges. In addition, three types of nodes are defined: decision nodes, corner nodes, and terminal nodes. A decision node is an intersection that connects hallways. A corner node is where orientation is changed. A terminal node is the end of a hallway.

7.2 Indoor modeling with iLOC

In contrast, iLOC defines the navigation as traversing on the network of the POI graph. This graph is already constructed to be as close to the human approach of describing routes as possible. In order to achieve this, one has to come up with a methodological approach.

These are some of the lessons learned during the modeling work:

- Some buildings are built in a linear fashion, meaning that there are only a few routes inside. Other buildings might follow a more open design, allowing multiple options to reach the desired destination. Classifying the building before any actual modeling work will help to identify the right strategy.

- Larger indoor spaces should be divided up to smaller ones, where one is able to see all POIs nearby, which is crucial in the process of following an indoor route.
- Rooms having multiple connections to nearby rooms (e.g. hallways) should have a central POI acting as a hub, which will provide the connections to the surrounding rooms. These hubs are acting as major decision points in the traversal. In some cases, a secondary hub can be also very practical. Entrances to Vertical Passages are good POI-hub candidates.
- Staircases connect different floors, but sometimes large auditoriums also have doors to different floors. In this sense, the floor level of a room can be ambiguous.

The indoor navigation RDF data of the Óbuda University was generated manually since only the maps of the building floors in jpg form were available as the base of the work. First, the BuildingPart (Room and Floor) class entities were recorded, like labs, lecture halls, hallways, stairways, elevators, and floors. The relevant Room entities are connected by the *iloc:isPartOf* property with the relevant Floor entities, except for the Vertical Passage entities, which are are also connected by the iloc:isPartOf property with the relevant Building entity.

The following excerpt describes the Building and some BuildingPart entities. In the description of the data besides the iLOC ontology, the geo, vcard and rdfs are also used.

```
prefix : <http://lod.nik.uni-obuda.hu/data/> .
prefix iloc: <http://lod.nik.uni-obuda.hu/iloc/iloc#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
```

```
prefix geo:<http://www.w3.org/2003/01/geo/wgs84_pos#> .
prefix vcard: <http://www.w3.org/2006/vcard/ns#> .
:360a a iloc:Room ;
      iloc:isPartOf :Third_floor_oe_building .

:E70 a iloc:Elevator ;
      iloc:isPartOf :OE_Main_Building .

:Office_307 a iloc:Room ;
      rdfs:label "307 iroda"@hu ;
      iloc:isPartOf :Third_floor_oe_building .

:OE_Main_Building a iloc:Building ;
      rdfs:label "OE Main Building"@en ;
      geo:location :AddressOE .

:AddressOE a geo:SpatialThing , vcard:Work ;
      vcard:postal-code "1034" ;
      vcard:street-address "Bécsi út 96/B" ;
      vcard:hasLocality :Budapest ;
      vcard:hasCountryName :Hungary .
```

The next step was the POI network planning and implementation. The POI network is a graph with POI vertices. There is an edge between two POIs if the route between them can appear in the navigation. In the process, first, the POI entities were drawn and connected with each other by hand on the map. The complexity of this task required to work out a methodology to achieve the goal. In the following the main steps of the POI network modeling are described:

- POIs should exist at Room entrances, at stairway and elevator exits, at building entrances. In most cases, there should be a POI at each stairway exit. Exceptions might occur at short stairways, where only one POI might be enough.

- When defining Hallway entities, the larger hallway spaces should be divided up to smaller ones in such a way, that one is able to see all POIs of a certain Hallway entity nearby. It means, that a route between two POIs of the same Hallway entity is taken for granted, so it will not be computed. This point ensures that the computed route does not contain obvious route descriptions, so is not unnecessarily long. Figure 11 shows the result of this step of the process for the third-floor hallway, which is divided into four parts (identified with 360a, 360b1, 360b2, 360c on the floorplan diagram).

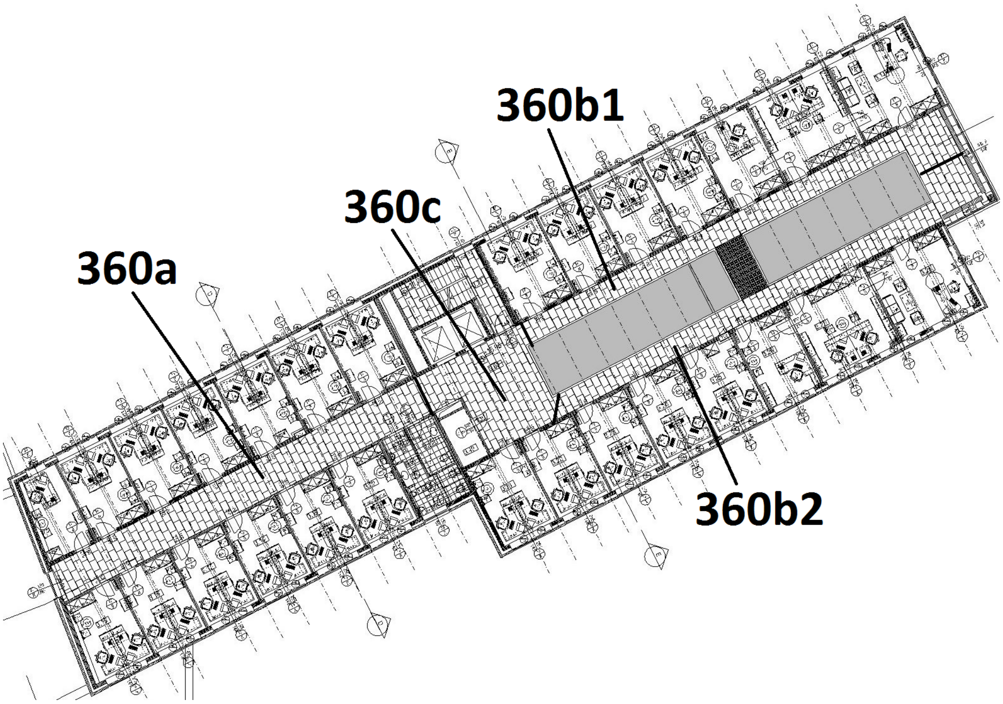


Figure 11: Hallway entities on the third floor

- In each Hallway entity, there should be a central POI acting as a POI-hub, which will provide the connections to the POIs of the surrounding rooms in the same Hallway entity. In the graph, the POI-hub is connected with all POIs belonging to the same Hallway entity. In some cases, a secondary hub can be also very practical. Entrances to Vertical

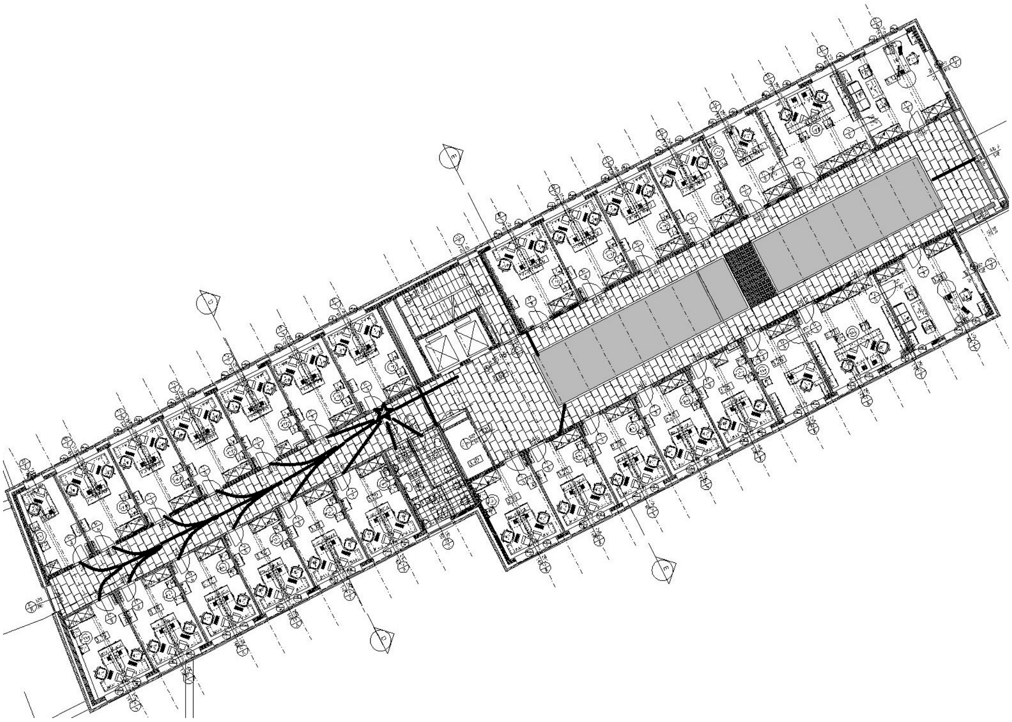


Figure 12: POI-hub in Hallway entity 360a

Passages are good POI-hub candidates. Figure 12 shows the POI-hub structure in Hallway class entity 360a.

- The POI-hubs are acting as major decision points in the traversal. When planning the POI network first the POI-hubs should be connected with each other. Then if necessary, some other POI-POI connection should be also included in the POI network. This step is illustrated in Figure 13.
- Staircases connect different floors, but sometimes large auditoriums also have doors to different floors. In this sense, the floor level of a room can be ambiguous.

- A stairway or elevator exit POI should be connected with all other exit POIs belonging to the same Stairway or Elevator entity. This optimizes the route generation time in the navigation process.

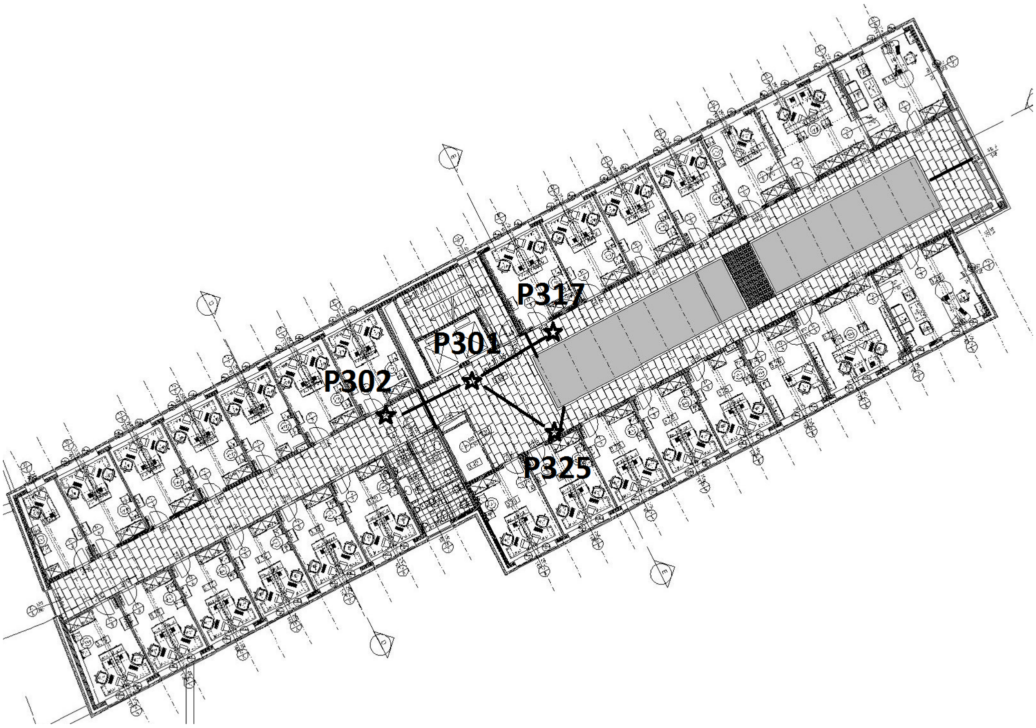


Figure 13: Connecting POI-hubs on the third floor

The following excerpt describes the P302 hub-POI instance, which belongs to Office 307 and has a photo assigned to it with the foaf:depiction property.

```
@prefix : <http://lod.nik.uni-obuda.hu/data/>
@prefix iloc: <http://lod.nik.uni-obuda.hu/iloc#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
:P302 a iloc:POI ;
      rdfs:label "Office_307 entrance"@en , "307 iroda
bejárata"@hu ;
      rdfs:comment "Starpoint" ;
      iloc:belongsToRoom :360a , :Office_307 ;
```

```
      iloc:connectsPOI :P301 , :P303 , :P304 , :P305 , :P306 ,
:P307 , :P308 , :P309 , :P310 , :P311 , :P312 , :P313 , :P314
, :P315 , :P316 ;
      foaf:depiction <http://lod.nik.uni-
obuda.hu/indoor/POI/Office_307.jpg>.
```

7.3 The Óbuda University use case

As a pilot project, the John von Neumann Faculty of Informatics building of the Óbuda University was modeled with iLOC and published as Linked Open Data.

The data can be reached from the SPARQL endpoint of Óbuda University (http://lod.nik.uni-obuda.hu/marmotta/). The data is organized in the named graph http://lod.nik.uni-obuda.hu/graphs/indoor-locations. The resulted dataset contains 1755 RDF triples. The LOD server hosting the indoor navigation data of the Óbuda University runs Apache Marmotta[29]. During the project, experiments were made also with Apache Fuseki[30].

The data for evaluation was created manually based on building layout diagrams of the Óbuda University (OU). The dataset was extended with links to the GeoNames[31] geographic dataset, which can provide the transition from outdoor to indoor navigation. The validity of the indoor location data and its conformance to the ontology were tested with Protégé.

Figure 14 shows screenshots of a mobile indoor navigation application developed for the university [27]. The application consumes linked open data

---

[29] http://marmotta.apache.org

[30] https://jena.apache.org/documentation/fuseki2/index.html

[31] http://www.geonames.org/

by accessing a SPARQL endpoint and issuing SPARQL queries to the university's LOD server[32], similarly to the example query in chapter 7.2.

The application is designed to show a POI based navigation route from a given starting point to a specific destination point. The result is displayed in two different ways; either as a list of POIs in one page representing the route, or the route is shown in consecutive pages, each showing the description of the next POI with an optional photo.

Útvonal áttekintése

Indulás: 306 iroda

Érkezés: F01 előadó

7/1

1. 306 iroda bejárata
2. 307 iroda bejárata
3. Személylift harmadik emeleti bejárata
4. Személylift földszinti bejárata
5. F02 előadó bejárata
6. Doberdó utcai bejárathoz vezető lépcső földszinti bejárata

Figure 14: Screenshot of a mobile indoor navigation application developed for the Óbuda University

---

[32] http://lod.nik.uni-obuda.hu

On the screenshot on Figure 14, a walking route is described (in Hungarian) between Office 306 on the third floor and the Lecture Hall F01 on the ground floor using the elevator. As the Óbuda University publishes timetable and other information as linked open data, the application is able to navigate to a person's office, or to the location of a specific course event.

7.4 Patient navigation in hospitals

The second use case, iLOC was evaluated for is patient navigation in hospitals. Hospitals are usually complex public places with thousands of visitors each year. Majority of the visitiors have very limited knowledge about the building or the campus, or have disabilities requiring special routes to move, but they still need to easily find a person or a department.

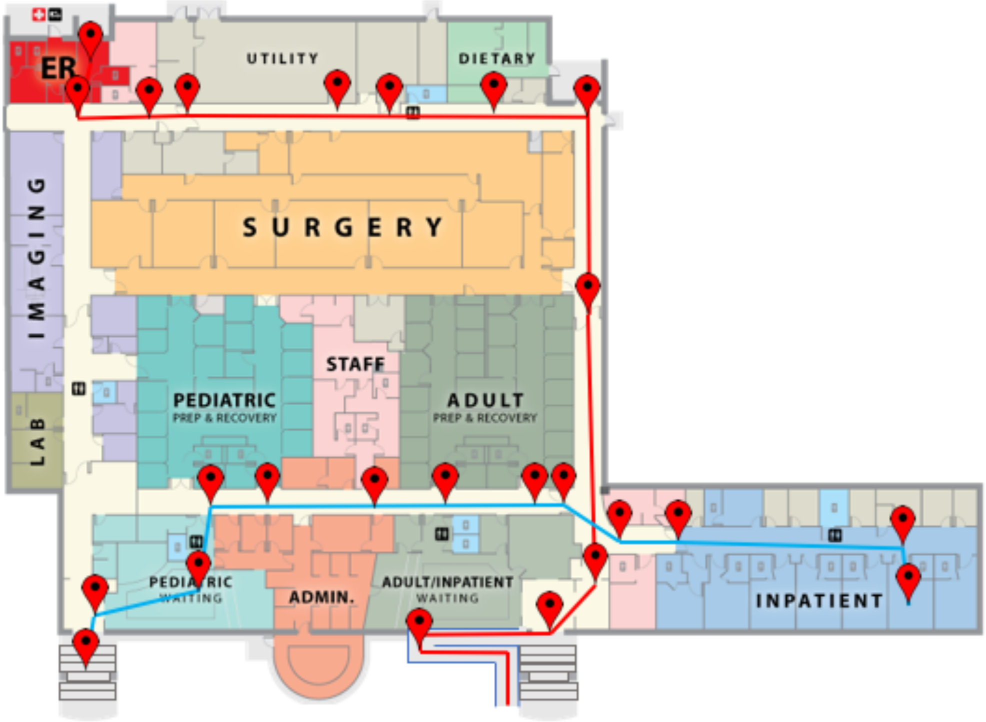


Figure 15: Example routes modelled within a smart hospital

A somewhat more complicated situation arises when the visitor has incomplete information about their goal. A good example for this is an inpatient with a relative visiting after her surgery. Since a smart hospital is offering mobile applications, in which every inpatient can share her status to their relatives, who will receive these updates as notifications. Once the visitors enter to the hospital, the app will navigate them to the right room, without knowing what the exact room number is or where the room is located within the building. In Figure 15, the blue route illustrates such an example.

People requiring accessible routes within the hospital building are especially needing publicly available information advising them about suitable routes. For example, a person using a wheelchair generally needs elevators between floors, wide-enough doors to cross, and avoid stairs, high door thresholds, and steep slopes. But different kinds of wheelchairs (e.g. an electric wheelchair) can travel through passages of different steepness, and a wheelchair user can overcome some smaller barriers with help. An elderly person may only look for the minimization of the number of stairs, or the distance to walk. Figure 15 illustrates a wheelchair accessible route, marked with red, from one building entrance to the emergency department, which leads through a ramp instead of steps. The ER front-desk is modelled as the default location of the ER department, so one does not need to know the exact room number to find it.

To support these needs, iLOC can be integrated with a domain ontology [24] describing the different accessibility options as well as the built-in properties can be used to provide generic route information: whether a route includes any steps or slopes. Also, with specialized SPARQL queries, the maximal distance one can walk can be considered and some lengthy paths can be excluded.

7.5 The Budapest Airport Terminal use case

For further validation, the ontology was tested with a third use case, focusing on how to model restrictions within a public building. As airports are great examples for complex restrictions, the public area of Terminal 2 of Budapest Airport was chosen to validate iLOC. We created a high-level model representing the major parts of the airport terminal. During the validation, we recognized the need for modelling one-way passages, because on an airport there are routes where one cannot go back on the same way (e.g. security screening).

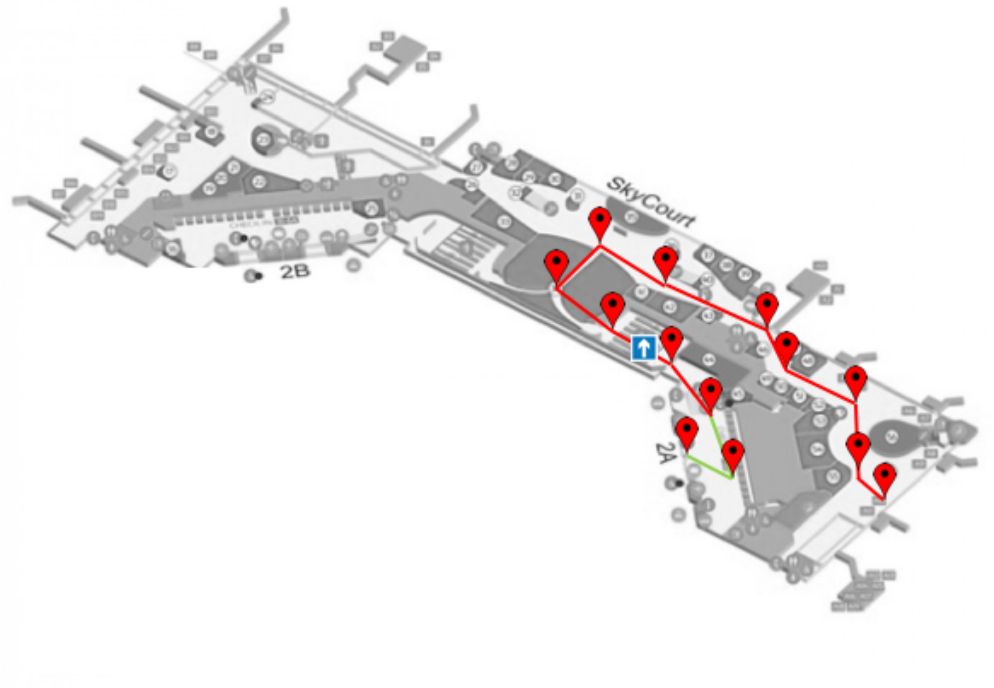


Figure 16: A route modeled on the Budapest Airport Terminal 2A

Figure 16 illustrates a route from one of the entrances of the Terminal 2A through the check-in counters and security to the gate A10. The route leads through 13 POIs and has an open and restricted part (marked with green and red lines), where one needs a valid boarding pass to enter. One section of the route is one-way only, marked with the standard white arrow on a blue background symbol. The airside was divided into smaller, easy to describe spaces, to allow smooth route finding.

7.6 Conclusion

In this chapter, indoor modeling best practices were shown. These were identified during multiple trials working with the iLOC ontology. For indoor navigation, one needs a network of easy to identify points within the indoor space. Using the floorplan, one needs to systematically identify these points, making sure there is an ideal coverage within the rooms. Smaller rooms with single entrance need to be treated differently than bigger rooms with multiple entrances. Also, the role of the room impacts the type and the number of the points to be identified. Next to identifying the right points on the floorplan, there are other special cases, where passages might be one-way or restricted by other rules. Also, to consider the personal needs and abilities require further data to be collected about the passages. The quality of the resulted semantic map relies heavily on the process of the map's production.

8 Summary

In this chapter, we conclude this thesis by reflecting on the research topics described in section 1.1. For each of these topics, we recall the results from the chapters that addressed it, draw conclusions, and discuss related issues. We close with a general discussion and recommendations for future work in the domain of university related linked data publishing.

8.1 Summary of the work

This work is focusing on information sharing and consumption approaches enabling universities to act as a 'smart university', satisfying the new requirements and expectations of the 21th century.

Chapter 3 to 7 provide a summary of related work, highlighting some potential gaps and oppportunities to focus on. Related to Linked Open Data publishing in the university domain, we have identified the need for simplified management of complex data definitions. As with the previously available ontologies the representation of education related concepts is incomplete for our requirements (e.g. missing Curriculum concept and many relationships among the defined concepts), we needed an ontology providing full coverage on the major concepts in the Hungarian higher education. We also needed an ontology being able to describe campus related concepts and facilitating in-door navigation to enable step-by-step indoor navigation and also a methodology to model indoor spaces using that ontology.

8.1.1 Self-unfolding URIs

Linked Data resources are identified by Uniform Resource Identifiers. It is an important step in any Linked Data project to define the conventions for URI assignments. In some cases, resources already have their natural identifiers, or they can be inherited from previous databases, the challenge being to avoid duplicated keys. However, there are cases when frequent triple sets have to be added to the graph without any convenient and easy way to determine the concepts' identifier, leaving the only option to assign a random URI. In our paper [67] we elaborated on a mechanism that makes handling complex and frequent data points easier, and also provides the benefits of simple data authoring with richer querying or reasoning on the data. We demonstrated how to eliminate some of the time consuming and error-prone aspects of Linked Data authoring by introducing the self-unfolding URI concept. This solution generates RDF description to entities based on information encoded in their URIs. For the generation of these new RDF triples we proposed templates that can be implemented by SPARQL Insert queries.

8.1.2 OLOUD

The Ontology for Linked Open University Data (OLOUD) is a practical approach to model course information at a typical Hungarian university. We provided in the papers [22], [65] and [64] further details about the Ontology, which aims to integrate data from several sources and provide personal timetables, navigation and other types of help for students and lecturers. The modeled domains include curricula, subjects, courses, semesters and personnel, and also buildings and events. Although there are several ontologies for the listed domains, selecting a set of ontologies fitting our use case was not an easy task. We summarized our challenges, such as missing links, inconsistencies as well as many overlaps between ontologies. OLOUD acts as a glue for a selection of existing ontologies, and thus enabled us to formulate SPARQL queries for a wide range of practical questions of university students.

8.1.3 iLOC

The iLOC ontology supports publishing data about indoor locations as Linked Open Datasets, which enables indoor navigation if proper semantic models are built from the structural and geometric descriptions of the involved buildings using the ontology concepts. Currently, we do not know about any indoor navigation services utilizing Linked Data, and there is also a lack of online available vocabularies or ontologies for the description of building plans. Our papers [23], [24], [25] and [66] present the iLOC ontology as a generic approach for indoor wayfinding and location description. The suggested Linked Data model enables the integration of various datasets related to buildings and can be used not only in the university environment, but in museums, shopping malls, hospitals, campuses, stadiums, airports, etc. Possible use cases include the description of nearby places, planning routes or finding places with given features, even in emergency situations. iLOC supports three levels-of-detail for a building plan, with increasing complexity and capabilities. Furthermore, routes can be calculated with flexible constraints considering the users' goals and abilities. The ontology is demonstrated with a smartphone application helping university students to find their lectures. We were the first to demonstrate, that using standard SPARQL queries and the iLOC ontology, one can get personalized indoor routes from Linked (Open) Data.

Furthermore, in the paper [23] we are describing a methodology to create optimal models of indoor spaces dividing them into a network of POIs and finding the most optimal routes between these points. And also, a case study is provided about modeling of the building of the Óbuda University.

8.2 Future work

Future opportunities include enabling linked data publishing for more and more institutes, so data publishing systems based on the OLOUD ontology can act as a central hub for online services and mobile applications helping students and lecturers. We expect, that using our university linked data model, new services will appear with a wide range of support for the university life ranging from navigation and timetables to the discovery of research information and their key people at the university. The major benefit using ontologies, like iLOC, relies in the opportunity of using a shared data model describing public places and providing navigation services on the top of the data. A toolset is necessary to improve the practical value of the iLOC ontology. However, the majority of the tooling is to be done as future work.

8.3 Conclusion

With using these results, one is now capable to create and publish university related data as Linked Data, which can satisfy the needs of multiple audiences. The Linked Data approach enables the integration of most university data sources, now usually stored in separate databases. If an institute is able to provide such data for their users, it enables a whole ecosystem of mobile and web application developers and service providers building services on the top of this data. If multiple institutes decide on using these methods and ontologies, such applications and services will be able to consume data from multiple sources, which might enable new, even more interesting use cases.

We demonstrated the value by implementing one of the iLOC use cases with a dedicated mobile application. However, the indoor space modeling process and the linked data creation was carried out manually, which can and should be partially or fully automatized for future use cases.

9 Összefoglalás

Ez a fejezet az 1.1 fejezetben ismertetett kutatási témákat foglalja össze. Ismerteti mindegyik téma eredményét és a kapcsolódó kérdéseket. Végezetül a felsőoktatással kapcsolatos kapcsolt adatok publikálásának általános összegzésére és további lehetőségeinek bemutatására kerül sor.

9.1 Kutatási eredmények

A disszertáció azt a kérdést vizsgálja, hogy a megfelelő információ megosztásával hogyan tudnak a 21. században az egyetemek lépést tartani az új igényekkel, egyfajta “okos egyetemként” működve.

A 3-7. fejezetek bemutatják az adott terület hátterét, jelentősebb eredményeit és rávilágítanak különböző lehetőségekre az adott területen belül. A felsőoktatás szakterületét leíró kapcsolt nyílt adatok publikálásával kapcsolatban a következő észrevételeket tettük: szükséges bonyolult szerkezetű adatok egyszerű kezelhetőségét biztosítani. Valamint szükséges egy ontológia a magyar felsőoktatás fogalmainak a leírásához, mivel a már elérhetőek nem biztosítanak teljes lefedettséget (pl. a Tanterv fogalma, illetve sok különböző fogalmakat összekötő tulajdonságok hiányoznak), illetve szükséges egy ontológia a különböző (egyetemi) kampuszokkal kapcsolatos fogalmak leírására illetve beltéri navigáció megvalósítására valamint egy beltér modellezési módszerre, amely segítségével beltereket lehet leírni az ontológia fogalmait felhasználva.

9.1.1 Kibontható URI-k

A Kapcsolt Adathalmazok erőforrásait egységes erőforrás azonosítók, URI-k segítségével azonosítjuk. Fontos lépés minden Kapcsolt Adathalmazokkal foglalkozó munka során definiálni az URI hozzárendelések szabályait. Ugyan széles körben elfogadott vélekedés, hogy az URI-k ne hordozzanak semmilyen jelentést, a gyakorlatban hasznosnak bizonyul, ha ember által is olvashatóak, így egy hosszabb listából az emberi szem is ki tudja választani a keresett erőforráshoz tartozó URI-t.

Ha az erőforrásoknak természetes módon adódik azonosítója, vagy ezen azonosítók már valamilyen úton adottak, akkor a kihívás az ismétlődő azonosítók elkerülése. Ha az erőforrások nem rendelkeznek egyértelmű inverz-funkcionális tulajdonsággal, ami egyszerű azonosítási lehetőséget adna, csak összetett kulccsal lehet az ilyen jellegű erőforrásokat azonosítani. Ilyen erőforrásokra jó példa az ismétlődő események sorozata, ahol az egyes események nincsenek egymástól jól megkülönböztetve, így ezeket csak a pontos időpontjuk és helyszínük alapján lehet azonosítani. Az RDF (Resource Description Framework) lehetőséget ad ilyen esetekben az azonosítók teljes elhagyására Blank Node-ok használatával, de ez további problémákat okozhat, mivel a létrehozott erőforrások közvetlenül nem hivatkozhatók.

Emberi felhasználásra – különösen a kézi adatbevitel támogatására – kiemelkedően fontos, hogy valamilyen szemantikát adjunk az URI tartalmának. Ugyanakkor ez redundanciát okoz, hiszen az URI-ba kódolt információnak kapcsolódó entitások formájában is meg kell jelennie. Ha a redundancia nem elkerülhető, valamilyen automatikus mechanizmusnak biztosítani kell az adatok ellenőrzését és/vagy karbantartását.

A Kapcsolt Adatok feldolgozásához és főként a megfelelő lekérdezések végrehajtásához szükséges lehet az erőforrások teljes szemantikus leírása. A Kibontható URI-k ezen esetek kezelésére javasoltak: az ilyen URI-k egy előre meghatározott mintát követnek, tartozik hozzájuk egy vagy több sablon, ami leírja a struktúrájukat.

Sok olyan eset létezik, amikor gyakori RDF triplet-halmazokat kell létrehozni mindenféle egyszerű módon definiálható azonosítók nélkül. Ilyen esetben az egyik megoldás egy véletlenszerűen hozzárendelt azonosító, míg az általunk publikált [67] másik módszer ezeknek az eseteknek a kezelését egyszerűsíti le, illetve további sokrétű lekérdezési és következtetési lehetőséget ad, standard SPARQL felhasználásával.

Tegyük fel, hogy adott egy Kapcsolt Adathalmaz és egy mechanizmus (pl. trigger), ami egy adott típusú új egyed megjelenése esetén további RDF hármasokat ad az adathalmazhoz, amelyek szemantikusan az eredetiből származnak, ahhoz kapcsolódnak. Ez a mechanizmus leegyszerűsíti az RDF tárolók karbantartását azok lényegi módosítása nélkül.

Négy típuspéldán keresztül demonstráltuk az automatikusan kibontható URI-k felhasználási lehetőségeit:

1. összetett kulccsal azonosítható egyedek
2. Blank Node (anonim erőforrás) jelöltek
3. URI alapján történő kapcsolódó egyedek összekapcsolása
4. következtetés URI-ban megjelenő információk segítségével

Fontos része a kibontható URI-k körüli folyamatoknak a különböző sablonok, amelyek azokat a szabályokat reprezentálják, amelyek eldöntik, hogy milyen új információ származtatható az adott egyed URI-jából. Ezek a sablonok az adatokon történő lekérdezések konzisztenciáját hivatottak biztosítani. Egy sablon egy URI mintát és a hozzá tartozó generált RDF leírás struktúráját adja meg.

A javasolt URI séma a Mátrix URI-n alapul, amely képes több egymástól független paraméter segítségével azonosítani az erőforrást. Követve a Mátrix URI sémát, az ilyen URI az alap résszel kezdődik, amelyet az erőforrás típusa követ, majd az összetett kulcsának elemei kerülnek felsorolásra.

9.1.2 OLOUD

Az Ontology for Linked Open University Data (OLOUD) egy gyakorlati megközelítése a kurzusokkal kapcsolatos információk modellezésének a magyar felsőoktatásban. Cikkeinkben ([22], [64] és [65]) további részleteket közöltünk az ontológiáról, ami célul tűzte ki a különböző forrásokból származó adatok integrálásával a személyre szabott órarendek, navigáció és egyéb szolgáltatások biztosítását hallgatók és oktatók számára. Többek között a következő tárgyterületeket modelleztük: tanterv, tantárgy, kurzus, szemeszter, személyzet, épületek és események. A tárgyterület összetettsége miatt egyetlen ontológia túlságosan bonyolult lenne az összes felhasználási eset lefedésére. Több ontológia is létezik a tárgyterület egyes részeinek a leírására, így kiválasztottuk azokat, amelyek együtt ki tudják elégíteni az összes követelményünket. Összefoglaltuk a különböző kihívásokat és a rájuk adott válaszokat, mint például hiányzó kapcsolatok fogalmak között, inkonzisztens vagy egymással redundáns, átfedésben lévő ontológiák. A munka eredményeként az ontológia felhasznál és integrál különböző forrásokból származó fogalmakat, támogatva a személyes időbeosztással, navigációval, illetve egyéb szükségletekkel kapcsolatos adatok fogyasztását és publikálását. Az OLOUD ontológia ragasztóként köt össze és egészít ki meglévő ontológiákat, így egy egységes modellt ad a magyar felsőoktatás fontos fogalmainak a leírására és lehetővé teszi a széles körű SPARQL lekérdezésekkel megfogalmazott egyetemi igények kielégítését Kapcsolt Adatokkal.

9.1.3 iLOC

Napjainkban a kültéri navigáció széleskörben elterjedt szolgáltatás, ami nagy segítségünkre van autóvezetés, séta vagy akár a tömegközlekedés használata során is. Ez a fajta navigáció általában a GPS vagy a Glonass rendszer jeleit használva koordinátákban adja meg a pozíciót. Épületek belsejében ugyanakkor egy navigációs rendszernek sokkal bonyolultabb útvonalakkal kell dolgoznia, amelyek folyosókon, felvonókon, illetve lépcsőkön vezetnek keresztül, miközben általában a GPS vagy Glonass jelek nem érhetőek el. A felhasználók korlátozottak lehetnek képességeiket, illetve hozzáférési jogosultságaikat illetően, valamint személyes preferenciáik sokkal széleskörűbbek lehetnek. A jelenleg elérhető megoldások általában valamilyen speciális és drága hardveres megoldáson alapulnak, amely segítségével a pontos hely megállapítható ugyan, de ellentétben a kültéri navigációval, nincsen egy elterjedt alapvető navigációs módszer.

Ebből az okból hoztuk létre az iLOC ontológiát, amely a belterekben történő navigációt támogatja. Az ontológia belterek leírását és annak Kapcsolt Nyílt Adathalmazként történő publikálását teszi lehetővé. Ilyen adatok felhasználásával beltéri navigációs szolgáltatás készíthető, ha az ontológia fogalmait felhasználva az épületekről szemantikus modell készül azok strukturális és geometrikus leírása (pl. alaprajzuk) alapján. Lehetővé teszi azt is, hogy két pont között különböző megszorításoknak is megfelelő (pl. kerekesszékkel járható) útvonalat, illetve annak hosszát is kiszámoljuk. Jelenleg nem ismert más beltéri navigációt biztosító szolgáltatás, amely Kapcsolt Adatok felhasználásával működik és nincsen olyan elérhető ontológia sem, amellyel ilyen módon leírhatók lennének épületek. Más, hasonló célt megvalósító megoldások alapvetően különböznek a modellezés költségében, illetve a felhasználás jellegében, valamint a felhasznált fogalmakban.

Cikkeinkben ([23], [24], [25] és [66]) bemutattuk az iLOC ontológiát és egy általános módszert beltéri útkeresésre és helyszínek leírására. A javasolt Kapcsolt Adatmodell lehetővé teszi többféle épületekhez kapcsolódó adathalmaz összekapcsolását és nem csak egyetemi környezetben használható, hanem sikerrel alkalmazható más nyilvános helyszínek, mint múzeumok, bevásárló központok, kórházak, stadionok, valamint repülőterek leírására is.

Az iLOC ontológiával megvalósított épületleírások más ontológiákkal történő integrálása a navigációs lekérdezések különböző kiegészítését teszi lehetővé ezen ontológiák fogalmainak felhasználásával. Ilyen kiegészítés lehet a helyiségek kategorizálása, illetve az útvonalak hozzáférési lehetőségeinek, illetve megszorításainak a leírása. A felhasználási lehetőségek között szerepel közeli helyek leírása és megtalálása azok tulajdonságai alapján, útvonal tervezés és navigálás akár vészhelyzeti körülmények között is. Az ontológia három különböző részletességű épület leírási lehetőséget biztosít, amelyek egyre növekvő részletességgel és pontossággal írják le egyre több részletét az épületnek. Worboys [76] osztályozása alapján az iLOC egy hibrid ontológia, ami támogatja a részek, a köztük lévő kapcsolatok és távolságuk leírását is. Az alkalmazott topológiai leírás a legpontosabb a lépésről-lépésre történő navigációhoz, amely kiegészíthető szemantikus leírással (pl. a helyiség típusa, a hozzáférhetőség módja).

Az Óbudai Egyetem Informatika Karának épületét modellezve mutattuk meg az ontológia fogalmaival történő modellezés lehetőségeit. Az így létrejött adathalmazt egy diplomamunka keretében erre a célra fejlesztett mobil alkalmazás segítségével használtuk fel és lehetővé tettük többek között, hogy az egyetemi hallgatók megtalálják az óráiknak a helyszíneit.

Elsőként mutattuk meg, hogy standard SPARQL lekérdezések és az iLOC ontológia segítségével személyre szabott útvonalak számíthatók ki Kapcsolt (Nyílt) Adathalmazokon.

Cikkünkben [23] közzétettünk egy módszertant is olyan beltéri modellek kialakítására, amelyekkel belterek ismert pontjai (POI) kapcsolhatók hálózatba és ezen pontok között optimális útvonalak számíthatók.

9.1.4 További lehetőségek

Az OLOUD ontológián alapuló adatpublikáló rendszerek központi szerepet tölthetnek be az oktatókat és hallgatókat támogató különböző online szolgáltatások és mobil alkalmazások között. Arra számítunk, hogy az általunk kidolgozott modellt használva új szolgáltatások jelennek meg széleskörűen támogatva az egyetemi élet különböző területeit a navigációtól az órarenden keresztül akár a kutatási témák és kutatók könnyebb megtalálásáig.

A legnagyobb előnye az olyan ontológiák használatának, mint az iLOC az, hogy egységes modellt ad nyilvános helyek leírására és az azokban történő navigációra. Ugyanakkor szükségesnek érezzük további eszközöknek a megjelenését, amelyek támogatják a modellezést és a navigációt. Az Óbudai Egyetem épületének szemantikus modelljét kézzel hoztuk létre, amely folyamatot a jövőben érdemes részben vagy egészben automatizálni.

9.1.5 Konklúzió

Ezen eredmények felhasználásával lehetőség nyílik a magyar felsőoktatással kapcsolatos Kapcsolt Adatok létrehozására és publikálására, amely adatok ezután többféle fogyasztó közönség igényeit ki tudják elégíteni. A megközelítés lehetővé tesz a legtöbb egyetemi adatforrás integrálását, amelyek jelenleg különálló rendszerekben és adatbázisokban léteznek.

Ha egy intézmény képes ilyen adatok publikálására, az megnyitja az utat külső fejlesztőknek teljesen új mobil és web alkalmazások készítésére, amelyek az adatok felhasználásával újabb lehetőségeket teremtenek. Ha több intézmény dönt úgy, hogy azonos alapokon, azonos ontológiákat felhasználva publikálja nyilvános adatait, akkor ezek az új mobil és web alkalmazások képesek lesznek akár több intézmény adatait is fogyasztani, ezáltal még érdekesebb felhasználási lehetőségeket nyitva. Ezt az lehetőséget demonstrálja a navigációs mobil alkalmazás is.

Appendix A.

This section discusses SPARQL query efficiency in route calculation using the iLOC ontology. The queries being used for route calculation have an upper limit for the maximal length of routes measured in steps. Following the scheme of the example query, which has 3 steps as the upper limit, one can create new queries for different upper limits.

The two most popular open source licensed triple stores were used in the measurements, according to DB-Engines Ranking[33].

```
PREFIX iloc: <http://lod.nik.uni-obuda.hu/iloc#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?distance ?sl ?l1 ?l2 ?l3 ?el WHERE {
BIND (<room1> AS ?start ).
BIND (<room2> AS ?end).
OPTIONAL {?start rdfs:label ?sl.}
OPTIONAL {?p1 rdfs:label ?l1.}
OPTIONAL {?p2 rdfs:label ?l2.}
OPTIONAL {?p3 rdfs:label ?l3.}
OPTIONAL {?end rdfs:label ?el.}
?p1 iloc:belongsToRoom ?start.
?p1 iloc:connectsPOI ?p2.
?p2 iloc:connectsPOI ?p3.
?plast iloc:belongsToRoom ?end.
FILTER (?p3 = ?plast || ?p2 = ?plast || ?p1 = ?plast )
BIND (if( ?p3 = ?plast , 3, if( ?p2 = ?plast , 2, if( ?p1 =
?plast , 1, -1))) AS ?distance)
} ORDER BY ?distance LIMIT 1
```

---

[33] https://db-engines.com/en/ranking/rdf+store

This example can find routes with up to 3 steps. The measurements were conducted on a Macbook Pro with 2.7Ghz i5 CPU, using Fuseki 3.10[34] with OWLFBRuleReasoner inference. Inference was necessary to make sure all inverse properties were properly calculated. The test dataset consisted of 5489 triples. Figure 17 summarizes the results.

Next, the measurements were repeated using the same data and query set and the same hardware, running on Virtuoso Open Source Edition v7.2. Figure 18 summarizes these results. While Fuseki was only using a single core and single thread, Virtuoso was able to utilize the full CPU, which explains some of the performance differences.

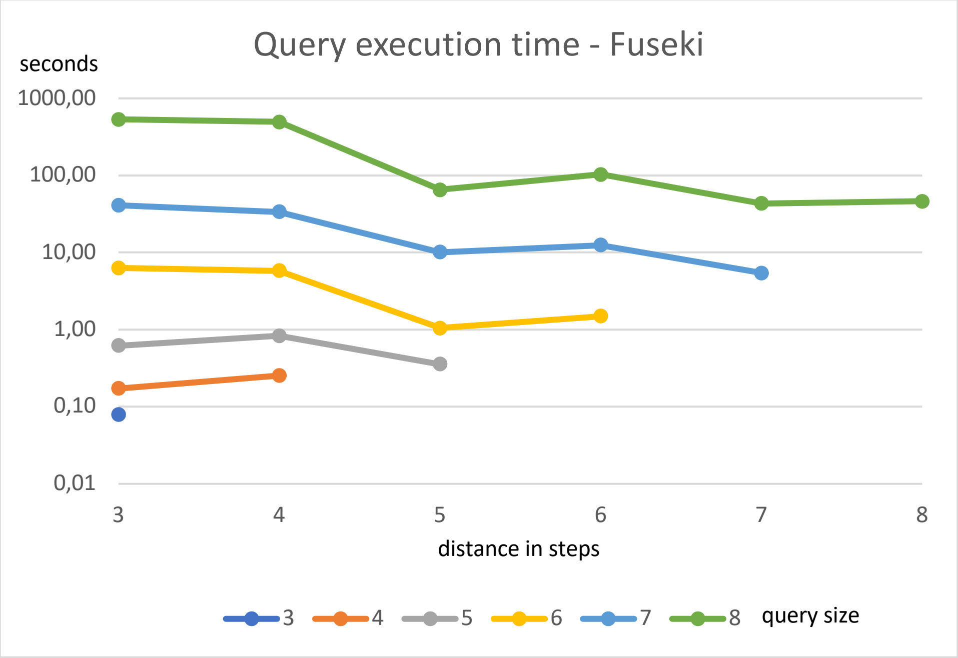


Figure 17: Average query times with Fuseki

---

[34] https://jena.apache.org/documentation/fuseki2/

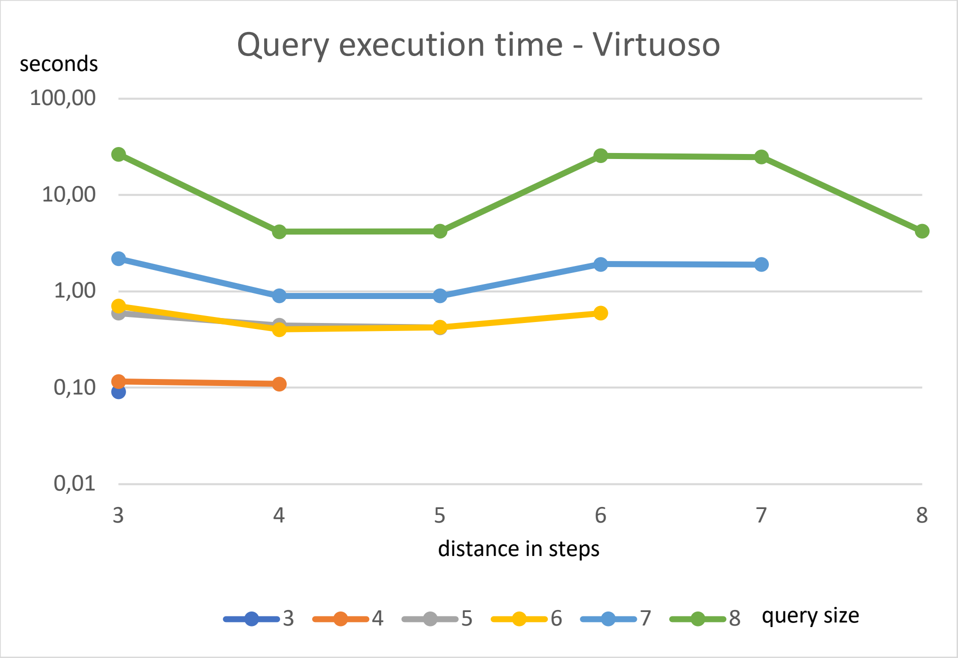


Figure 18: Average query times with Virtuoso

The best strategy to identify the shortest route is executing the queries in the order of their distance upper limit, as longer queries will be very inefficient for finding shorter routes.

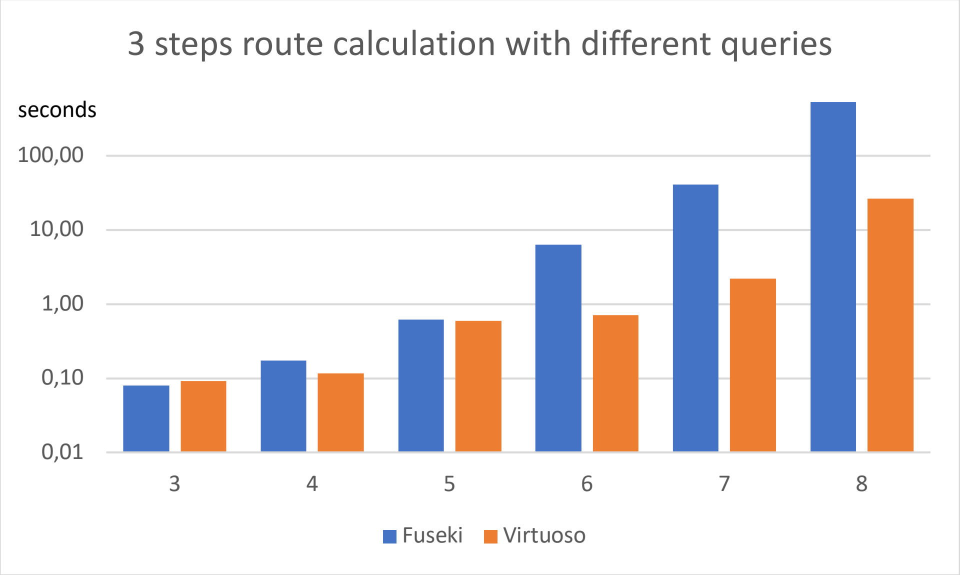


Figure 19: Average query time of 3 steps route calculation with different queries

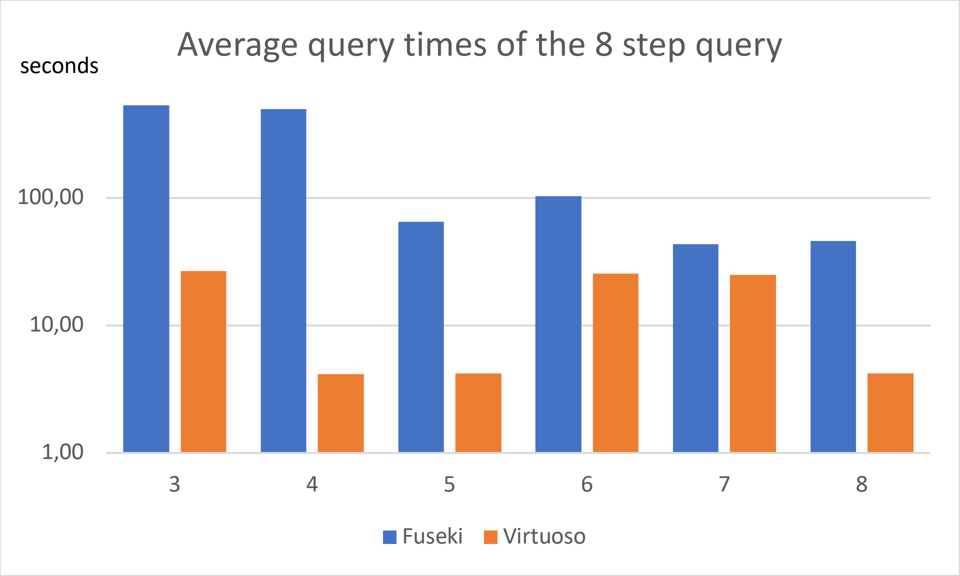


Figure 20: Average query times of the 8-step query

Figure 19 shows how longer queries get slower while finding a short, 3 steps distance. Figure 20 summarizes the efficiency of a longer query, finding different distances. Virtuoso was significantly more effective even with longer queries.

Table 4 and 5 compare the query execution times with different distances and query lengths. We conducted each measurement 3 times, table 6 and 7 shows the standard deviation in the query times.

Distance in steps	Query size					
	3	4	5	6	7	8
3	0.08 s	0.17 s	0.62 s	6.32 s	41.07 s	535.44 s
4		0.25 s	0.83 s	5.80 s	33.72 s	494.51 s
5			0.35 s	1.05 s	10.13 s	65.22 s
6				1.49 s	12.46 s	103.35 s
7					5.45 s	43.20 s
8						46.00 s

Table 4: Average query times – Fuseki

Distance in steps	Query size					
	3	4	5	6	7	8
3	0.09 s	0.12 s	0.59 s	0.7 s	2.19 s	26.57 s
4		0.11 s	0.45 s	0.4 s	0.9 s	4.18 s
5			0.42 s	0.42 s	0.9 s	4.21 s
6				0.6 s	1.92 s	25.6 s
7					1.9 s	24.86 s
8						4.21 s

Table 5: Average query times - Virtuoso

Distance in steps	Query size					
	3	4	5	6	7	8
3	0.02	0.04	0.23	0.19	2.31	12.21
4		0.09	0.29	0.55	0.46	8.45
5			0.11	0.25	0.13	0.77
6				0.12	0.12	3.39
7					0.03	0.44
8						1.19

Table 6: Standard deviations in the query times – Fuseki

Distance in steps	Query size					
	3	4	5	6	7	8
3	0.03	0.02	0.11	0.09	0.11	0.03
4		0.01	0.06	0.05	0.06	0.15
5			0.06	0.06	0.05	0.12
6				0.06	0.05	0.3
7					0.06	0.31
8						0.18

Table 7: Standard deviations in the query times – Virtuoso

Appendix B.

SPARQL examples

The following SPARQL queries demonstrate the capabilities of the OLOUD and iLOC ontologies. Using FILTERs results can be limited to specific entities (rooms, courses, etc.), as the examples below return all data.

1. What are the courses for a given subject in a specific semester?

```
PREFIX oloud:<http://lod.nik.uni-obuda.hu/oloud/oloud#>
SELECT DISTINCT ?course WHERE {
  ?course a <http://purl.org/vocab/aiiso/schema#Course> .
  ?course oloud:courseSubject <http://lod.nik.uni-
obuda.hu/data/NAIAB0SAND> .
  ?course oloud:courseTerm <http://lod.nik.uni-
obuda.hu/data/2014Fall> .
}
```

2. On which day a course is, what time does it start and how long it is?

```
PREFIX oloud: <http://lod.nik.uni-obuda.hu/oloud/oloud#>
PREFIX ta: <http://ontology.ihmc.us/temporalAggregates.owl#>
PREFIX time: <http://www.w3.org/2006/time#>
SELECT DISTINCT ?course ?day ?beginhour ?beginminute
?durationhour ?durationminute {
  ?course a <http://purl.org/vocab/aiiso/schema#Course> .
  ?course oloud:courseTime ?ct .
  ?ct ta:hasTemporalAggregateDescription ?tad .
  ?tad ta:hasithTemporalUnit ?day ;
       ta:hasStart ?start .
  ?start time:hasDurationDescription ?dd ;
         time:hasBeginning ?begin .
  ?dd time:hours ?durationhour ;
      time:minutes ?durationminute .
  ?begin time:inDateTime ?begindatetime .
  ?begindatetime time:hour ?beginhour ;
                 time:minute ?beginminute . }
```

3. What courses are at the same time partially or completely overlapping?

```
PREFIX oloud: <http://lod.nik.uni-obuda.hu/oloud/oloud#>
PREFIX ta: <http://ontology.ihmc.us/temporalAggregates.owl#>
PREFIX time: <http://www.w3.org/2006/time#>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?course1 ?course2
{
?course1 oloud:courseTime ?ct1.
?ct1 ta:hasTemporalAggregateDescription ?ta1.
?ta1 ta:hasithTemporalUnit ?day;
        ta:hasStart ?s1.
?s1 time:hasBeginning ?begin1 ;
    time:hasDurationDescription ?duration1.
?begin1 time:inDateTime  ?dt1 .
?dt1 time:hour ?hour1 ;
    time:minute ?minute1 .
?duration1 time:hours ?durationHour1;
 time:minutes ?durationMinute1.
?course2 oloud:courseTime ?ct2.
?ct2 ta:hasTemporalAggregateDescription ?ta2.
?ta2 ta:hasithTemporalUnit ?day;
        ta:hasStart ?s2.
?s2 time:hasBeginning ?begin2 .
?begin2 time:inDateTime  ?dt2 .
?dt2 time:hour ?hour2 ;
    time:minute ?minute2 .
BIND     (xsd:integer(?minute1)+(xsd:integer(?hour1)*60)     AS
?start1).
BIND     (xsd:integer(?minute2)+(xsd:integer(?hour2)*60)     AS
?start2).
BIND
(xsd:integer(?durationMinute1)+(xsd:integer(?durationHour1)*60
)+?start1 AS ?end1).
FILTER (?start1 < ?start2 && ?start2 < ?end1).
FILTER (?course1 != ?course2 ).
}
```

4. What is the course schedule (with course identifier, time and lecturer) for a specific lecture hall or lab?

```
PREFIX oloud: <http://lod.nik.uni-obuda.hu/oloud/oloud#>
PREFIX ta: <http://ontology.ihmc.us/temporalAggregates.owl#>
```

```
PREFIX time: <http://www.w3.org/2006/time#>
PREFIX iloc: <http://lod.nik.uni-obuda.hu/iloc/iloc#>
SELECT  DISTINCT  ?course  ?room  ?day  ?beginhour  ?beginminute
?teacher WHERE {
  ?room a iloc:Room.
  ?course oloud:locatedAt ?room;
          oloud:courseTeacher ?teacher;
          oloud:courseTime ?ct .
  ?ct ta:hasTemporalAggregateDescription ?tad .
  ?tad ta:hasithTemporalUnit ?day ;
       ta:hasStart ?start .
  ?start time:hasDurationDescription ?dd ;
         time:hasBeginning ?begin .
  ?dd time:hours ?durationhour ;
      time:minutes ?durationminute .
  ?begin time:inDateTime ?begindatetime .
  ?begindatetime time:hour ?beginhour ;
                 time:minute ?beginminute .
}
```

5. What are the required courses of a specific course?

```
PREFIX oloud: <http://lod.nik.uni-obuda.hu/oloud/oloud#>
SELECT DISTINCT ?dep WHERE {
  <http://lod.nik.uni-obuda.hu/data/NAIDR0SAND>
oloud:subjectRequires ?dep
}
```

6. Navigate from the main entrance to a specific course location!

```
PREFIX oloud: <http://lod.nik.uni-obuda.hu/oloud/oloud#>
PREFIX iloc: <http://lod.nik.uni-obuda.hu/iloc/iloc#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?distance ?sl ?l1 ?l2 ?l3 ?el WHERE {
<http://lod.nik.uni-obuda.hu/data/AB0_EA_2014-15-1>
oloud:locatedAt ?end
BIND (<http://lod.nik.uni-obuda.hu/data/PB01> AS ?start ).
OPTIONAL {?start rdfs:label ?sl.}
OPTIONAL {?p1 rdfs:label ?l1.}
OPTIONAL {?p2 rdfs:label ?l2.}
OPTIONAL {?p3 rdfs:label ?l3.}
OPTIONAL {?end rdfs:label ?el.}
?start iloc:connectsPOI ?p1.
```

```
?p1 iloc:connectsPOI ?p2.
?p2 iloc:connectsPOI ?p3.
?plast iloc:belongsToRoom ?end.
FILTER (?p3 = ?plast || ?p2 = ?plast || ?p1 = ?plast )
BIND (if( ?p3 = ?plast , 3, if( ?p2 = ?plast , 2, if( ?p1 =
?plast , 1, -1))) AS ?distance)
}
ORDER BY ?distance LIMIT 1
```

7. What are the dates of the individual sessions of a given course?

```
PREFIX oloud: <http://lod.nik.uni-obuda.hu/oloud/oloud#>
PREFIX ta: <http://ontology.ihmc.us/temporalAggregates.owl#>
PREFIX time: <http://www.w3.org/2006/time#>
SELECT DISTINCT ?course ?day ?beginhour ?beginminute WHERE {
  ?course oloud:courseTime ?ct .
  ?ct ta:hasMember ?start .
  ?start time:hasDurationDescription ?dd ;
         time:hasBeginning ?begin .
  ?dd time:hours ?durationhour ;
      time:minutes ?durationminute .
  ?begin time:inDateTime ?begindatetime .
  ?begindatetime time:hour ?beginhour ;
                 time:minute ?beginminute .
}
```

8. What event is going to be at a specific location?

```
PREFIX oloud: <http://lod.nik.uni-obuda.hu/oloud/oloud#>
PREFIX ta: <http://ontology.ihmc.us/temporalAggregates.owl#>
PREFIX time: <http://www.w3.org/2006/time#>
PREFIX event: <http://purl.org/NET/c4dm/event.owl#>
SELECT DISTINCT ?event ?place ?day ?beginhour ?beginminute {
  ?event event:time ?t ;
         event:place ?place .
  ?t time:hasDurationDescription ?dd ;
         time:hasBeginning ?begin .
  ?dd time:hours ?durationhour ;
      time:minutes ?durationminute .
  ?begin time:inDateTime ?begindatetime .
  ?begindatetime time:hour ?beginhour ;
                 time:minute ?beginminute .
}
```

Appendix C.

Ontology for Indoor Navigation

IRI: http://lod.nik.uni-obuda.hu/iloc#

Date: 01/07/2015

Imported Ontologies:

http://qudt.org/1.1/schema/qudt#
http://www.w3.org/2006/vcard/ns

Classes

Building

IRI: http://lod.nik.uni-obuda.hu/iloc#Building
Represents a whole building. It may have an address and spatial coordinates.
has super-classes: Location

Building Entrance

IRI: http://lod.nik.uni-obuda.hu/iloc#BuildingEntrance
Represents the entrance where one can enter/leave the building.
has super-classes: Entrance

Building Part

IRI: http://lod.nik.uni-obuda.hu/iloc#BuildingPart
Superclass of every part of a building.
has super-classes: Location
has sub-classes: Floor, Room

Elevator

IRI: http://lod.nik.uni-obuda.hu/iloc#Elevator
Connects Rooms on different floors.
has super-classes: VerticalPassage

Entrance

IRI: http://lod.nik.uni-obuda.hu/iloc#Entrance
Represents any type of entrance
has super-classes: Point of Interest
has sub-classes: Building Entrance, Room Entrance

Floor

IRI: http://lod.nik.uni-obuda.hu/iloc#Floor
It is part of a building. It may consist of Rooms.
has super-classes: Building Part

Location

IRI: http://lod.nik.uni-obuda.hu/iloc#Location
Superclass for all location.
has super-classes: SpatialThing
has sub-classes: Building, Building Part, Point of Interest
is in domain of: is part of
is in range of: is part of

point

IRI: http://www.w3.org/2003/01/geo/wgs84_pos#Point
Uniquely identified by lat/long/alt. i.e.
spaciallyIntersects(P1, P2) :- lat(P1, LAT), long(P1, LONG), alt(P1, ALT), lat(P2, LAT), long(P2, LONG), alt(P2, ALT).
sameThing(P1, P2) :- type(P1, Point), type(P2, Point), spaciallyIntersects(P1, P2).
has super-classes: SpatialThing

Point of Interest

IRI: http://lod.nik.uni-obuda.hu/iloc#POI
An entity with Location. Can be used as part of a route.
has super-classes: Location
has sub-classes: Entrance
is in domain of: belongs to Room, connects to POI, connects to POI (one way)
is in range of: connects to POI, connects to POI (one way), connects to RouteSection, connects to RouteSection, connects to RouteSection, has POI

Room

IRI: http://lod.nik.uni-obuda.hu/iloc#Room
is defined by: http://yago-knowledge.org/resource/wordnet_room_104105893
Represents any type of room within a building.
has super-classes: Building Part
has sub-classes: VerticalPassage
is in domain of: has POI
is in range of: belongs to Room

Room Entrance

IRI: http://lod.nik.uni-obuda.hu/iloc#RoomEntrance
Represents the entrance where one can enter/leave the room.
has super-classes: Entrance

Route constraint

IRI: http://lod.nik.uni-obuda.hu/iloc#RouteConstraint
RouteConstraint instances represent any abilities or permissions required to traverse a Route Section. Such Route Constraints can be a required employee badge or the ability to climb stairs.
is in domain of: has Category

Route section

IRI: http://lod.nik.uni-obuda.hu/iloc#RouteSection
Route section directly connecting two POIs

is in domain of: connects to RouteSection, connects to RouteSection, connects to RouteSection
is in range of: has Category

Spatial Thing

IRI: http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing
Anything with spatial extent, i.e. size, shape, or position. e.g. people, places, bowling balls, as well as abstract areas like cubes.
has sub-classes: Location, point
is in domain of: altitude, latitude, longitude
is in range of: location

Stairway

IRI: http://lod.nik.uni-obuda.hu/iloc#Stairway
Connects Rooms on different floors.
has super-classes: VerticalPassage

Vertical Passage

IRI: http://lod.nik.uni-obuda.hu/iloc#VerticalPassage
Connects Rooms on different floors using any method.
has super-classes: Room
has sub-classes: Elevator, Stairway

Object Properties

belongs to Room

IRI: http://lod.nik.uni-obuda.hu/iloc#belongsToRoom
This property states that a Point of Interest belongs to a Room. POIs can belong to multiple rooms: e.g. doors are part of both rooms.
has domain: Point of Interest
has range: Room
is inverse of: has POI

connects to POI

IRI: http://lod.nik.uni-obuda.hu/iloc#connectsPOI
This property is used to describe a direct route between POIs.
has characteristics: symmetric
has super-properties: connects to POI (one way)
has domain: Point of Interest
has range: Point of Interest

connects to POI (one way)

IRI: http://lod.nik.uni-obuda.hu/iloc#connectsPOIOneWay
This property is used to describe a direct walkable one-way route between POIs. Only walking from the subject to the object is possible/allowed.
has characteristics: asymmetric
has sub-properties: connects to POI
has domain: Point of Interest
has range: Point of Interest

connects to RouteSection

IRI: http://lod.nik.uni-obuda.hu/iloc#routeFromPOI
This property is used to define the start POI of a RouteSection instance.
has super-properties: connects to RouteSection
has domain: Route section
has range: Point of Interest

connects to RouteSection

IRI: http://lod.nik.uni-obuda.hu/iloc#routeToFromPOI
This property is used to define either one of the POIs of a RouteSection instance.
has sub-properties: connects to RouteSection, connects to RouteSection
has domain: Route section
has range: Point of Interest

connects to RouteSection

IRI: http://lod.nik.uni-obuda.hu/iloc#routeToPOI
This property is used to define the target POI of a RouteSection instance.
has super-properties: connects to RouteSection
has domain: Route section
has range: Point of Interest

has Category

IRI: http://lod.nik.uni-obuda.hu/iloc#hasCategory
This property states that a Room or Point of Interest has a specific category, like Hallway, Office, etc. Category can be any resource.

has Constraint

IRI: http://lod.nik.uni-obuda.hu/iloc#hasConstraint
This property states that a RouteSection has a specific constraint, requiring permission or ability to traverse.
has domain: Route constraint
has range: Route section

has POI

IRI: http://lod.nik.uni-obuda.hu/iloc#hasPOI
This property states that there is a Point of Interest in a Room. Multiple rooms can include the same POI: e.g. doors are part of both rooms.
has domain: Room
has range: Point of Interest
is inverse of: belongs to Room

is part of

IRI: http://lod.nik.uni-obuda.hu/iloc#isPartOf
Object property to specify another Location which includes (is broader of) this Location.
has domain: Location
has range: Location

location

IRI: http://www.w3.org/2003/01/geo/wgs84_pos#location

The relation between something and the point, or other geometrical thing in space, where it is. For example, the relationship between a radio tower and a Point with a given lat and long. Or a relationship between a park and its outline as a closed arc of points, or a road and its location as an arc (a sequence of points). Clearly in practice there will be limit to the accuracy of any such statement, but one would expect an accuracy appropriate for the size of the object and uses such as mapping.
has range: SpatialThing
is also defined as: annotation property

Annotation Properties

altitude

IRI: http://www.w3.org/2003/01/geo/wgs84_pos#alt
The WGS84 altitude of a SpatialThing (decimal meters above the local reference ellipsoid).
has domain: SpatialThing

based near

IRI: http://xmlns.com/foaf/0.1/based_near
has sub-properties: location

date

IRI: http://purl.org/dc/elements/1.1/date

latitude

IRI: http://www.w3.org/2003/01/geo/wgs84_pos#lat
The WGS84 latitude of a SpatialThing (decimal degrees).
has domain: SpatialThing

location

IRI: http://www.w3.org/2003/01/geo/wgs84_pos#location
The relation between something and the point, or other geometrical thing in space, where it is. For example, the relationship between a radio tower and a Point with a given lat and long. Or a relationship between a park and its outline as a closed arc of points, or a road and its location as an arc (a sequence of points). Clearly in practice there will be limit to the accuracy of any such statement, but one would expect an accuracy appropriate for the size of the object and uses such as mapping.
has super-properties: based near
is also defined as: object property

longitude

IRI: http://www.w3.org/2003/01/geo/wgs84_pos#long
The WGS84 longitude of a SpatialThing (decimal degrees).
has domain: SpatialThing

term status

IRI: http://www.w3.org/2003/06/sw-vocab-status/ns#term_status

OLOUD: Ontology for Linked Open University Data

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#
Date: 09/06/2016
Imported Ontologies:

http://lod.nik.uni-obuda.hu/iloc/iloc#
http://lod.nik.uni-obuda.hu/oloud/oloud-time#
http://purl.org/NET/c4dm/event.owl
http://purl.org/vocab/aiiso/schema-20080925.rdf
http://www.w3.org/2006/time
http://xmlns.com/foaf/spec/20140114.rdf

Classes

Attendance Pattern

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#AttendancePattern
Type of the training, which can be full-time, part-time (at the evenings) or correspondence (at the weekends).
is in range of: attendance pattern
has members: correspondence, full time, part time

Course term

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#CourseTerm
Course term is a portion of the academic year, the time during which an educational institution holds classes.
has super-classes: interval
is in range of: course term

Course time

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#CourseTime
Course time represents the temporal sequence when a specific course is hold.
has super-classes: temporal seq
is in range of: course time

Course type

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#CourseType
Course type represents the different formats (e.g. lecture, seminar, laboratory or practice) of courses at universities.
is in range of: course type
has members: exam course, lab, lecture, practise

Curriculum

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#Curriculum
After enrolling to the university each student is assigned to a Curriculum, which is a set of Subjects and their relations (i.e. dependencies among the subjects) that a student has to accomplish.
has super-classes: knowledge grouping

is in domain of: end date, start date
is in range of: curriculum

Degree

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#Degree
An academic degree is the state of recognized completion of studies at a college or university. The most common degrees awarded today are bachelor's, master's and doctoral degrees.
is in range of: degree
has members: bsc, msc, phd

Specialization

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#Specialization
The specialization can be acquired as part of the Study Programme, providing special expertise in training, which is indicated on the proof of successful completion of the university diploma.
has super-classes: knowledge grouping
is in range of: specialization
has members: elective, obligatory part

Object Properties

attendance pattern

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#attendancePattern
This property shows the Attendance Pattern of the subject or curriculum.
has domain: knowledge grouping
has range: Attendance Pattern

course subject

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#courseSubject
This property connects the specific course with its subject.
has domain: course
has range: subject

course teacher

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#courseTeacher
This property connects the specific course with its teacher.
has domain: course
has range: person

course term

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#courseTerm
This property shows the course term when the specific course is hold.
has domain: course
has range: Course term

course time

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#courseTime
This property links a specific course to a temporal sequence of individual course events.

has super-properties: has Temporal Sequence
has domain: course
has range: Course time

course type

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#courseType
This property connects the specific course with its course type.
has domain: course
has range: Course type

curriculum

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#curriculum
This property shows the Curriculum where the Subject belongs to.
has domain: subject
has range: Curriculum

degree

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#degree
This property shows the academic degree of the subject or the curriculum.
has domain: knowledge grouping
has range: Degree

end date

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#endDate
This property shows the time when a curriculum becomes void.
has domain: Curriculum
has range: instant

has Temporal Sequence

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#hasTemporalSeq
This property links a recurring event to a temporal sequence of individual events.
has sub-properties: course time
has range: temporal seq

located at

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#locatedAt
This property shows the location of the event or the course.
has domain: event **or** course
has range: location

specialization

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#specialization
This property shows the Specialization that a specific Curriculum has or the Specialization of a Subject.
has domain: knowledge grouping
has range: Specialization

start date

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#startDate
This property shows the time when a curriculum comes into force.
has domain: Curriculum

has range: instant

study programme

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#studyProgramme
This property shows the Study Programme that the certain Subject or Curriculum belongs to.
has domain: knowledge grouping
has range: programme

subject requires

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#subjectRequires
This property shows the name of the subject which completion is necessary in order to start a course of the certain subject.
has domain: subject
has range: subject
is also defined as: data property

Data Properties

subject credit

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#subjectCredit
Subject credit is a unit that gives weight to the value, level or time requirements of the subject taken at the university.
has domain: subject
has range: int

subject lab number

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#subjectLabNo
The number of Laboratory units that a given Subject contains per week.
has domain: subject
has range: int

subject lecture number

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#subjectLectureNo
The number of Lecture units that a given Subject contains per week.
has domain: subject
has range: int

subject requires

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#subjectRequires
This property shows the name of the subject which completion is necessary in order to start a course of the certain subject.
has range: string
is also defined as: object property

Named Individuals

bsc

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#BSc
belongs to: Degree

computer engineer

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#ComputerEngineer
belongs to: programme

correspondence

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#Correspondance
belongs to: Attendance Pattern

elective

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#Elective
belongs to: Specialization

engineer teacher

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#EngineerTeacher
belongs to: programme

exam course

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#ExamCourse
belongs to: Course type

full time

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#FullTime
belongs to: Attendance Pattern

lab

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#Lab
belongs to: Course type

lecture

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#Lecture
belongs to: Course type

msc

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#MSc
belongs to: Degree

obligatory part

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#ObligatoryPart
belongs to: Specialization

part time

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#PartTime
belongs to: Attendance Pattern

phd

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#PhD
belongs to: Degree

practise

IRI: http://lod.nik.uni-obuda.hu/oloud/oloud#Practise
belongs to: Course type

Appendix D.

A List of papers of the author and citations to them

1) Szász, Barnabás, Rita Fleiner, and András Micsik. "iLOC–Building Indoor Navigation Services using Linked Data." *Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) 12-15 September 2016 Leipzig, Germany.* CEUR, 2016.
    a) Piros, Peter, Rita Fleiner, and Levente Kovács. "Linked data generation for courses and events at Óbuda University." *Applied Machine Intelligence and Informatics (SAMI), 2017 IEEE 15th International Symposium on*. IEEE, 2017. 253-258.
    b) Bajaj, Garvita, Rachit Agarwal, Pushpendra Singh, Nikolaos Georgantas and Valérie Issarny. "4W1H in IoT semantics." *IEEE Access* 6 (2018): 65488-65506.
    c) Chalhoub, Nidal, and Gabriella Simon-Nagy. "Indoor Navigation Based on Linked Data at Honvéd Hospital, Budapest." *2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE, 2018.
2) Fleiner, Rita, Barnabás Szász, and Péter Piros. "Indoor navigation Linked Data at Obuda University." *Applied Computational Intelligence and Informatics (SACI), 2016 IEEE 11th International Symposium on*. IEEE, 2016.
    a) Piros, Peter, Rita Fleiner, and Levente Kovács. "Linked data generation for courses and events at Óbuda University." *Applied Machine Intelligence and Informatics (SAMI), 2017 IEEE 15th International Symposium on*. IEEE, 2017. pp. 253-258.
    b) Chalhoub, Nidal, and Gabriella Simon-Nagy. "Indoor Navigation Based on Linked Data at Honvéd Hospital, Budapest." *2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE, 2018.

3) Fleiner, Rita, Barnabas Szasz, and Andras Micsik. "OLOUD-An Ontology for Linked Open University Data." *Acta Polytechnica Hungarica* 14.4. 2017. pp. 63-82.
   a) Nguyen, Viet Bach. "PhD Odyssey." Proceedings of the EKAW Doctoral Consortium 2018 co-located with the 21st International Conference on Knowledge Engineering and Knowledge Management (EKAW 2018). CEUR, 2018.
4) Szász, Barnabás, Rita Fleiner, and András Micsik. "A case study on Linked Data for University Courses." *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, Cham, 2016.
   a) Piros, Peter, Rita Fleiner, and Levente Kovács. "Linked data generation for courses and events at Óbuda University." *Applied Machine Intelligence and Informatics (SAMI), 2017 IEEE 15th International Symposium on*. IEEE, 2017. pp. 253-258.
   b) Nahhas, Safia, Omaimah Bamasag, Maher Khemakhem, and Nada Bajnaid. "Added Values of Linked Data in Education: A Survey and Roadmap." *Computers* 7, no. 3 (2018): pp. 45.
   c) Tapia-Leon, Mariela, Abdón Carrera Rivera, Janneth Chicaiza, and Sergio Luján-Mora. "Application of ontologies in higher education: A systematic mapping study." In *Global Engineering Education Conference (EDUCON), 2018 IEEE*, pp. 1344-1353.
5) Fleiner, Rita, Gabriella Simon-Nagy, and Barnabás Szász. "Accessible indoor navigation based on linked data in hospitals." *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, IEEE, 2016. pp. 2425-2430.
6) Fleiner, Rita, Barnabás Szász, Gabriella Simon-Nagy, and András Micsik. "Indoor navigation for motion disabled persons in medical facilities." *Acta Polytechnica Hungarica* 14, no. 1. 2017. pp. 111-128.
   a) Csaba, György, László Somlyai, and Zoltán Vámossy. "Mobil robot navigation using 2D LIDAR." In *2018 IEEE 16th world symposium on applied machine intelligence and informatics (SAMI)*, IEEE, 2018. pp. 000143-000148.
   b) Chalhoub, Nidal, and Gabriella Simon-Nagy. "Indoor Navigation Based on Linked Data at Honvéd Hospital, Budapest." *2018 IEEE 12th*

    International Symposium on Applied Computational Intelligence and Informatics (SACI). IEEE, 2018.
    c) Simon-Nagy, Gabriella, and Rita Fleiner. "Ontology Extension for Personalized Accessible Indoor Navigation." In *International Conference on Global Research and Education*, pp. 281-288. Springer, Cham, 2017.

7) Szász, Barnabás, Rita Fleiner, and András Micsik. "Linked data enrichment with self-unfolding URIs." *Applied Machine Intelligence and Informatics (SAMI), 2016 IEEE 14th International Symposium on*, pp. 305-309. IEEE, 2016.
    a) Piros, Peter, Rita Fleiner, and Levente Kovács. "Linked data generation for courses and events at Óbuda University." *Applied Machine Intelligence and Informatics (SAMI), 2017 IEEE 15th International Symposium on*. IEEE, 2017. pp. 253-258.
    b) Debattista, Jeremy, Christoph Lange, Sören Auer, and Dominic Cortis. "Evaluating the Quality of the LOD Cloud: An Empirical Investigation."
8) Szász, Barnabás, Rita Fleiner, and András Micsik *Practical uses of location and event data as Linked Open University Data.* FutureRFID 2014. Eger, Eszterházy Károly Főiskola. 2014. pp. 151-159.

References

[1] Abiteboul, Serge. "Querying semi-structured data." International Conference on Database Theory. Springer, Berlin, Heidelberg, 1997.

[2] Aggarwal, Charu C., and Haixun Wang. "Graph data management and mining: A survey of algorithms and applications." Managing and mining graph data. Springer, Boston, MA, 2010. pp. 13-68.

[3] Anagnostopoulos, Christos, Vassileios Tsetsos, and Panayotis Kikiras. "OntoNav: A semantic indoor navigation system." 1st Workshop on Semantics in Mobile Environments (SME05), Ayia. 2005.

[4] Baader, Franz, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider "The description logic handbook: Theory, implementation and applications." Cambridge university press, 2003.

[5] Baader, Franz, Ian Horrocks, and Ulrike Sattler. "Description logics." Foundations of Artificial Intelligence 3. 2008. pp. 135-179.

[6] Barker, Phil. "A short project on linking course data." 2015. [online] Available: http://blogs.pjjk.net/phil/a-short-project-on-linking-course-data/ [Accessed: 28-Jan-2019]

[7] Berners-Lee, Tim. "Linked Data-Design Issues." 2006. [online] Available: http://www.w3.org/DesignIssues/LinkedData.html [Accessed: 28-Jan-2019]

[8] Börner, Katy, Michael Conlon, Jon Corson-Rikert, and Ying Ding. "VIVO: A semantic approach to scholarly networking and discovery." Synthesis lectures on the Semantic Web: theory and technology 7, no. 1. 2012. pp. 1-178.

[9] Brickley, Dan, and Libby Miller. "FOAF Vocabulary Specification 0.99, Namespace Document 14 January 2014-Paddington Edition." 2014. [online] Available: http://xmlns.com/foaf/spec/ [Accessed: 28-Jan-2019]

[10] CEN. "Educational Credit Information Model (ECIM)" CEN WORKSHOP AGREEMENT. 2010.

[11] Corcho, Oscar, and Raúl García-Castro. "Five challenges for the semantic sensor web." Semantic Web 1.1, 2. 2010. pp. 121-125.

[12] Correndo, Gianluca, Manuel Salvadores, Ian Millard and Nigel Shadbolt "Linked timelines: Time representation and management in linked data." First International Workshop on Consuming Linked Data (COLD 2010), Shanghai, China. 2010.

[13] d'Aquin, Mathieu, Alessandro Adamou, and Stefan Dietze. "Assessing the educational linked data landscape." Proceedings of the 5th Annual ACM Web Science Conference, ACM, 2013, pp. 43-46.

[14] d'Aquin, Mathieu, The LUCERO Project, [online] Available: http://lucero-project.info/lb/2012/04/so-whats-in-linked-datasets-for-education/. [Accessed: 28-Jan-2019]

[15] D'Arcus, Bruce, and Frédérick Giasson. "Bibliographic ontology specification." Madrid: Biblioteca Nacional Española, 2009.

[16] Daga, Enrico, Mathieu d'Aquin, Alessandro Adamou, and Stuart Brown. "The open university linked data–data.open.ac.uk." Semantic Web 7, no. 2, 2016, pp. 183-191.

[17] DCMI Usage Board. DCMI Metadata Terms. DCMI, 2012, [online] Available: http://dublincore.org/documents/dcmi-terms/ [Accessed: 28-Jan-2019]

[18] Demartini, Gianluca, Iliya Enchev, Joël Gapany, and Philippe Cudré-Mauroux. "The Bowlogna ontology: Fostering open curricula and agile knowledge bases for Europe's higher education landscape." Semantic Web 4, no. 1. 2013. pp. 53-63.

[19] Dodds, Leigh, and Ian Davis. "Linked Data Patterns: A pattern catalogue for modeling, publishing, and consuming Linked Data, Linked Data Patterns." 2012. [online] Available: http://patterns.dataincubator.org/book/ [Accessed: 31-Jan-2019]

[20] Dudas, Patrick M., Mahsa Ghafourian, and Hassan A. Karimi. "ONALIN: Ontology and algorithm for indoor routing." Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on. IEEE, 2009. pp. 720-725

[21] Euzenat, Jérôme, and Pavel Shvaiko. Ontology matching. Vol. 18. Springer, Heidelberg. 2007.

[22] Fleiner, Rita, Barnabás Szász, and András Micsik. "OLOUD-An Ontology for Linked Open University Data." Acta Polytechnica Hungarica 14.4. 2017. pp. 63-82.

[23] Fleiner, Rita, Barnabás Szász, and Péter Piros. "Indoor navigation Linked Data at Obuda University." Applied Computational Intelligence and Informatics (SACI), 2016 IEEE 11th International Symposium on. IEEE, 2016.

[24] Fleiner, Rita, Barnabás Szász, Gabriella Simon-Nagy, and András Micsik. "Indoor navigation for motion disabled persons in medical facilities." Acta Polytechnica Hungarica 14, no. 1. 2017. pp. 111-128.

[25] Fleiner, Rita, Gabriella Simon-Nagy, and Barnabás Szász. "Accessible indoor navigation based on linked data in hospitals." Systems, Man, and

Cybernetics (SMC), 2016 IEEE International Conference on, IEEE, 2016. pp. 2425-2430.

[26] Gruber, Thomas R. "A translation approach to portable ontology specifications." Knowledge acquisition 5.2. 1993. pp. 199-220.

[27] Gyöngyösi, Tamás. "Mobile application based on linked Open University Data." BSc Thesis. Óbuda University. 2016.

[28] Haase, Peter, Tobias Mathäß, and Michael Ziller. "An evaluation of approaches to federated query processing over linked data." Proceedings of the 6th International Conference on Semantic Systems. ACM, 2010. p. 5.

[29] Hasan, Souleiman, Edward Curry, Mauricio Banduk, and Sean O'Riain. "Toward situation awareness for the semantic sensor web: Complex event processing with dynamic linked data enrichment." Proceedings of the 4th International Conference on Semantic Sensor Networks-Volume 839. CEUR-WS.org, 2011. pp. 69-81.

[30] Haslhofer, Bernhard, Elaheh Momeni, Manuel Gay, and Rainer Simon "Augmenting Europeana content with linked data resources." Proceedings of the 6th International Conference on Semantic Systems. ACM, 2010. pp. 40.

[31] He, Bin, Mitesh Patel, Zhen Zhang, and Kevin Chen-Chuan Chang. "Accessing the deep web." Communications of the ACM 50.5. 2007. pp. 94-101.

[32] Heath, Tom, and Christian Bizer. "Linked data: Evolving the web into a global data space." Synthesis lectures on the semantic web: theory and technology 1.1. 2011. pp. 1-136.

[33] Hitzler, Pascal, Markus Krötzsch, Bijan Parsia, Peter Patel-Schneider, and Sebastian Rudolph. "OWL 2 web ontology language primer." W3C recommendation 27.1 2009. 123.

[34] Hobbs, Jerry R., and Feng Pan. "Time ontology in OWL." W3C working draft 27. 2006: pp. 133.

[35] Horrocks, Ian, Oliver Kutz, and Ulrike Sattler. "The Even More Irresistible SROIQ." Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006, AAAIPress, 2006. pp. 57-67.

[36] Iannella, Renato, and James McKinney. "vCard Ontology-for describing People and Organizations." W3C Group Note. 2014. [online] Available: http://www.w3.org/TR/vcard-rdf/ [Accessed: 31-Jan-2019]

[37] Janowicz, Krzysztof, Hitzler, Pascal, Adams, Benjamin, Kolas, Dave, and Vardeman II, Charles. "Five stars of linked data vocabulary use." Semantic Web 5.3. 2014. pp. 173-176.

[38] Kauppinen, Tomi, Johannes Trame, and Andres Westermann." Teaching core vocabulary specification." LinkedScience.org, Technical Report, 2012.

[39] Kolbe, Thomas H., Gerhard Gröger, and Lutz Plümer. "CityGML: Interoperable access to 3D city models." Geo-information for disaster management. Springer, Berlin, Heidelberg, 2005. pp. 883-899.

[40] Ma, Yuanchao, Bin Xu, Yin Bai, and Zonghui Li. "Building linked open university data: Tsinghua university open data as a showcase." Joint International Semantic Technology Conference, pp. 385-393. Springer, Berlin, Heidelberg, 2011.

[41] Mallea, Alejandro, Marcelo Arenas, Aidan Hogan, and Axel Polleres. "On blank nodes." International Semantic Web Conference. Springer, Berlin, Heidelberg, 2011. pp. 421-437.

[42] Matuszka, Tamás, Gergő Gombos, and Attila Kiss. "A new approach for indoor navigation using semantic webtechnologies and augmented reality." International Conference on Virtual, Augmented and Mixed Reality. Springer, Berlin, Heidelberg, 2013. pp. 202-210.

[43] Michael Lutz. "ARE3NA D.TD.03 Guidelines on methodologies for the creation of RDF (final version)". Technical report. 2014.

[44] Micsik, András, Sándor Turbucz, and Attila Györök. "LODmilla: a Linked Data Browser for All." Proceedings of the Posters and Demos Track of 10th International Conference on Semantic Systems - SEMANTiCS2014. 2014. pp. 31-34.

[45] Miles, Alistair, and Sean Bechhofer. "SKOS simple knowledge organization system reference." W3C recommendation 18, W3C, 2009.

[46] MLO CEN/ISSS "WS-LT Metadata for Learning Opportunities (MLO)." 2008.

[47] Nagel, Claus, Jiyeong Lee, Ki-Joune Li, Sisi Zlatanova, Thomas H. Kolbe, and Thomas Becker. "OGC IndoorGML". Open Geospatial Consortium standard. 2014

[48] Pan, Feng. "Temporal aggregates for web services on the semantic web." Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on. IEEE, 2005.

[49] Parundekar, Rahul, Craig A. Knoblock, and José Luis Ambite. "Linking and building ontologies of linked data." International Semantic Web Conference. Springer, Berlin, Heidelberg, 2010. pp. 598-614.

[50] Polleres, Axel, Aidan Hogan, Renaud Delbru, and Jürgen Umbrich. "RDFS and OWL reasoning for linked data." Reasoning Web. Semantic Technologies for Intelligent Data Access. Springer, Berlin, Heidelberg, 2013. pp. 91-149.

[51] Poveda-Villalón, María, Mari Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. "Validating ontologies with oops!." International Conference on Knowledge Engineering and Knowledge Management. Springer, Berlin, Heidelberg, 2012.

[52] Rahm, Erhard, and Philip A. Bernstein. "A survey of approaches to automatic schema matching." the VLDB Journal 10.4. 2001. pp. 334-350.

[53] Raimond, Yves, and Samer A. Abdallah. "The event ontology," OWL-DL ontology, 2006." Technical report [online] Available: http://motools.sourceforge.net/event [Accessed: 28-Jan-2019]

[54] Rodriguez, Benedicto, and Millard, Ian. "ReSIST Courseware Ontology" 2006. [online] Available: http://courseware.rkbexplorer.com/ontologies/courseware [Accessed: 21-Jun-2017]

[55] Rodriguez, Marko A. "The gremlin graph traversal machine and language (invited talk)." Proceedings of the 15th Symposium on Database Programming Languages. ACM, 2015. pp. 1-10

[56] Rohs, Michael, and Jürgen Bohn. "Entry points into a smart campus environment-overview of the ETHOC system." Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on Distributed Computing Systems Workshops, 2003. pp. 260-266. IEEE, 2003.

[57] Schaible, Johann, Thomas Gottron, and Ansgar Scherp. "Survey on common strategies of vocabulary reuse in linked open data modeling." European Semantic Web Conference, Springer, Cham, 2014. pp. 457-472.

[58] Sequeda, Juan F., and Oscar Corcho. "Linked stream data: A position paper." 2009.

[59] Shvaiko, Jérôme, and Euzenat Pavel. "Ontology Matching" Springer-Verlag New York. Inc. 2007.

[60] Speicher, Steve, John Arwe, and Ashok Malhotra. "Linked data platform 1.0." W3C Recommendation, February 26, 2015.

[61] Stubbs, Mark. "XCRI | eXchanging Course-Related Information." Technical report, Manchester Metropolitan University, Aytoun Street, Manchester. 2006.

[62] Styles, Rob, and Nadeem Shabir. "Academic institution internal structure ontology (AIISO)." 2008. [online] Available: http://vocab.org/aiiso/schema [Accessed: 28-Jan-2019]

[63] Styles, Rob, Chris Wallace, and Knud Moeller. "Participation ontology." 2008. [online] Available: http://vocab.org/participation/schema [Accessed: 28-Jan-2019]

[64] Szász, Barnabás, Rita Fleiner, and András Micsik. "Practical uses of location and event data as Linked Open University Data." FutureRFID 2014. Eger, Eszterházy Károly Főiskola. 2014. pp. 151-159.

[65] Szász, Barnabás, Rita Fleiner, and András Micsik. "A case study on Linked Data for University Courses." OTM Confederated International Conferences" On the Move to Meaningful Internet Systems. Springer, Cham, 2016.

[66] Szász, Barnabás, Rita Fleiner, and András Micsik. "iLOC–Building Indoor Navigation Services using Linked Data." Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) 12-15 September 2016 Leipzig, Germany. CEUR, 2016.

[67] Szász, Barnabás, Rita Fleiner, and András Micsik. "Linked data enrichment with self-unfolding URIs." Applied Machine Intelligence and Informatics (SAMI), 2016 IEEE 14th International Symposium on, pp. 305-309. IEEE, 2016.

[68] Szász, Barnabás, Rita Fleiner, and András Micsik. "OLOUD – Ontology for Linked Open University Data." 2016. [online] Available: http://lod.nik.uni-obuda.hu/oloud/index.html [Accessed: 28-Jan-2019]

[69] Szász, Barnabás, Rita Fleiner, and András Micsik. "OLOUD: Ontology for Linked Open University Data." 2016. [online] Available: http://lod.nik.uni-obuda.hu/oloud/oloud-20160609.owl [Accessed: 28-Jan-2019]

[70] Szász, Barnabás, Rita Fleiner, and András Micsik. "Ontology extension for Temporal data." 2016. [online] Available: http://lod.nik.uni-obuda.hu/oloud/oloud-time-20160609.owl [Accessed: 28-Jan-2019]

[71] Szász, Barnabás, Rita Fleiner, and András Micsik. Linked Data Enrichment with Self-Unfolding URIs – examples, 2016. [online] Available: http://lod.nik.uni-obuda.hu/unfolding/example.html [Accessed: 28-Jan-2019]

[72] Szász, Barnabás, Rita Fleiner, András Micsik and Gabriella, Simon-Nagy. "iLOC – An Indoor Ontology." 2016. [online] Available:

http://lod.nik.uni-obuda.hu/iloc/iloc-20160409.owl [Accessed: 28-Jan-2019]

[73] Uschold, Michael, "Building ontologies: Towards a unified methodology." Technical report. University of Edinburgh Artifical Intelligence Applications Intiture AIAI TR. 1996.

[74] Uschold, Michael, and Martin King. "Towards a methodology for building ontologies." IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing. 1995.

[75] Vandenbussche, Pierre-Yves, and Bernard Vatant. "Metadata recommendations for linked open data vocabularies." Version 1. 2011-12.

[76] Worboys, Michael. "Modeling indoor space." Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness. ACM, 2011. pp. 1-6.