Automated Ontology Matching in the Architecture, Engineering and Construction Domain - A Case Study

Georg Ferdinand Schneider[1,2][0000−0002−2033−859X]

[1] Fraunhofer Institute for Building Physics IBP, Fürther Straße 250, 90429 Nürnberg, Germany
[2] Technische Hochschule Nürnberg, Fürther Straße 250, 90429 Nürnberg, Germany
georg.schneider@ibp.fraunhofer.de

Abstract. The ontology-based modelling of the built environment is deemed promising to successfully integrate disparate knowledge silos and has gained significant attraction in industry and academia. This interest has led to a proliferating number of ontologies and the manual definition of schema level alignments among them is a tedious task. Hence, this paper explores the possibilities of automated ontology matching methods in this regard. This work compares manually created and automatically generated alignments of six domain ontologies to the building topology ontology. The initial findings of this case study indicate that current state of the art ontology matching tools are in principle capable of detecting automatically correct alignments and that their is a strong need to define domain specific benchmarks.

Keywords: Automated Ontology Matching · Architecture · Engineering · Construction · Facility Management · Heterogeneity · Building Topology Ontology.

1 Introduction

Ontology-based modelling and associated implementations based on Semantic Web Technologies (SWT) [10] have gained attention by academia and industry in the Architecture, Engineering, Construction and Facility Management (AEC/FM) domain [23]. A main motivation to use the technology is its ability to successfully address the problem of integrating heterogeneous information silos distributed across the AEC/FM domain [2].

The spread of the technology has lead to a proliferating development of domain ontologies (cf. reviews in [7,23,25]). This development poses the risk of putting the benefits of the technology at stake as the defined ontologies overlap and found ontological design patterns are reimplemented again and again making a reuse difficult. 'Thus, merely using ontologies, like using XML, does not reduce heterogeneity: it raises heterogeneity problems to a higher level.' [12]. The principle of ontology reuse stipulated in well-known ontology engineering methodologies [19,14] is frequently not followed, when designing domain ontologies in the building domain. This makes it cumbersome for potential developers to implement applications as again a heterogeneous landscape of domain models appears.

The Building Topology Ontology (BOT), initially defined in [25,26] and further developed by the members of the W3C Linked Building Data Community Group (W3C LBD CG) [32], has been proposed to define commonly reoccurring design patterns in domain ontologies of the AEC/FM domain. These design patterns then can be reused by developers in their respective domains through extending from BOT [29]. Following this approach, intrinsically a domain wide interoperability can be ensured.

The successful manual alignment of five domain ontologies (SAREF4Building [24], BRICK [3,4], DogOnt [6], ThinkHome [27], ifcOWL [22]) to BOT is presented in an initial effort in Schneider [29]. The schema level alignment of ontologies is a tedious task, which potentially is as challenging as ontology engineering itself [12]. There exists a strong need to use automated ontology matching methods to automatically find alignments [7]. Also ontologies tend to evolve over time and alignments need to be updated and checked accordingly. The analysis of the current state of the art presented in section 2 indicates that the use of automated ontology matching methods has not been studied in depth in the AEC/FM domain so far.

The contributions of this paper are two fold. First, manual alignments originally defined in an earlier contribution [29] are revised and updated to reflect the latest version of BOT (v0.3.0). Second, a study is conducted, where an automated ontology matching method [13] is utilised to align the respective domain ontologies to BOT. The generated alignments are compared to the manually found ones.

The remainder of this paper includes a review of existing work (Section 2) related to the automated matching of ontologies in the AEC/FM domain. Then in Section 3 a methodology is presented to compare manually defined and automated generated alignments of domain ontologies. Finally, in Section 4 revised manual alignments are presented and the results of the comparison to automatically generated alignments are presented in Section 5.

2 Related Work

The field of ontology matching has been around for a number of years and the fundamentals of the technology are presented in Euzenat & Shvaiko [12,30]. Specific matching algorithms are developed actively and their performance is evaluated yearly in benchmark tests under the supervision of the Ontology Alignment Evaluation Initiative (OAEI), where the results of the most recent event are presented in Algergawy *et al.* [1].

A, yet limited, number of contributions, which investigate the topic of ontology matching in the context of the AEC/FM domain exist.

In their demand for interoperability in the smart cities domain, Costin & Eastman [7] conduct a thorough review of existing contributions in this regard. They conclude the ontology-based modelling of the domain based on SWT provide means to address the prevalent heterogeneity. However, as the manual alignment of domain ontologies is a tedious task, the demand for automated ontology matching methods is made.

Otero-Cerdeira *et al.* [20] investigate the use of automated ontology matching methods in the context of smart cities. The present *OntoPhil* a ontology matching technique specifically designed for the matching of disparate knowledge sources in the context of smart cities. Beside the cited works no further documentation or download possibility of the OntoPhil tool has been found.

Bellini *et al.* [5] present a system for the integration of disparate data sources in the context of smart cities. The system is designed to handle large data volumes and integrates them by mapping the data to the Knowledge Model for City (KM4City) ontology. The actual mapping is undertaken manually using the Karma Data Integration tool [15].

Gyrard *et al.* [16] present an approach to enrich ontology catalogues with domain ontologies of smart cities. Their approach aims for interoperability among applications by providing an interface, the catalogue, to developers to easily find and reuse existing domain ontologies. An automated update is discussed 'but a manual checking is preferred to handle synonyms' [16].

Espinoza-Arias *et al.* [11] review existing ontological representations of smart city data. No explicit mappings are defined among the ontologies but after a characterisation a number of reoccurring ontology design patterns are defined.

The presented contributions indicate that automated ontology matching methods seem to be promising to address heterogeneity of formats and formal models. Most work reviewed focuses on ontology matching and alignment in the domain of smart cities. The AEC/FM domain can be seen as a sub-domain of smart cities but it has not been discussed in detail to the best of the authors knowledge.

3 Methodology

A methodology is established to evaluate and compare the manual and automated matching of ontologies in the AEC/FM domain. In the study the definition of alignments between BOT [25,26] and six domain ontologies is studied (SAREF4Building [24], BRICK [3,4], DogOnt [6], ThinkHome [27], ifcOWL [22], DERI Room [8]). The namespaces used in the work are reported in Table 1. In particular the following steps are conducted:

1. Manual definition of alignments on class and object property level;
2. Use of the AgreementMakerLight [13] tool to automatically generate alignments;
3. Comparison of the obtained manual and automatically created alignments.

The manual definition of alignments is an extension and revision of the work documented in an earlier contribution [29]. The step involves the retrieval and local storage of the most recent version of all involved ontologies. The definition of an alignment ontology which performs a full import (`owl:import`) of BOT and the respective domain ontology. Finally, alignments are defined manually through the use of the Protégé ontology editor [18]. The defined alignments mainly use subsumption for alignment. This has been found beneficial in discussion within the W3C LBD CG as the semantic implied by subsumption are less rigid as compared the definition of equivalences. Equivalence (e.g. `owl:equivalenceClass`) implies that all statements on one class also are true for the other, which might not always be the case. An OWL DL reasoner is invoked on the resulting alignment ontology (Pellet [31]) to ensure the consistency. The defined ontologies are published in an online repository[3].

A large number of tools and associated algorithms exist to perform automated ontology matching [12,21,1]. An an initial attempt here the *AgreementMakerLight* tool [13] is chosen from an extensive list of available tools[4]. The usage of the AgreementMakerLight tool has been found intuitive and the tool is available open-source in as a compiled java library from a web repository. The tool is used with default settings and BOT is always used as the source ontology.

Table 1. Namespaces and prefixes used in this work.

Prefix	Value
rdf	`http://www.w3.org/1999/02/22-rdf-syntax-ns#`
rdfs	`http://www.w3.org/2000/01/rdf-schema#`
owl	`http://www.w3.org/2002/07/owl#`
bot	`https://w3id.org/bot#`
s4bldg	`https://w3id.org/def/saref4bldg#`
dog	`http://elite.polito.it/ontologies/dogont.owl#`
th	`https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/BuildingOntology.owl#`
thvs	`https://www.auto.tuwien.ac.at/downloads/thinkhome/ontology/BuildingOntologySharedVocabulary.owl#`
ifc	`http://www.buildingsmart-tech.org/ifcOWL/IFC4_ADD2#`
brick	`https://brickschema.org/schema/1.0.2/Brick#`
rooms	`http://vocab.deri.ie/rooms#`

4 Manually Defined Alignments

In this section the results of the extended, revised and manually defined alignments between BOT ontology and other domain ontologies are reported.

[3] `https://github.com/w3c-lbd-cg/bot/tree/AlignmentRevision`, Last accessed: 20 May 2019

[4] `http://www.mkbergman.com/2129/30-active-ontology-alignment-tools/`, Last accessed: 20 May 2019

4.1 SAREF Extension for Building Devices

The SAREF Extension for Building Devices (SAREF4Building) [24] ontology is an ontology to extend the SAREF ontology [9] into the buildings domain. A description of the ontology can be found in its documentation and reviews [24,23,29].

The defined alignments are reported in Table 2 and the rationale behind is described in the following paragraph.

SAREF4Building describes the concepts of `s4bldg:Building`s and `s4bldg:-Space`s, which can be defined as specialisations of `bot:Building` and `bot:Space`. As the focus of SAREF and its extension to the buildings domain focus on the description of tangible devices, `s4bldg:PhysicalObject`s, `s4bldg:Sensor`s and `s4bldg:Actuator`s qualify as `bot:Element`s, which again has been formalised through subsumption.

The, compared to the last iteration of alignments [29], newly introduced high level relationships `bot:containsZone` and `bot:containsElement` reflect on a high level the semantics of `s4bldg:hasSpace` and `s4bldg:contains` object properties of SAREF4Building and are aligned through defining them as subproperties of the respective BOT object properties (see Table 2).

Table 2. Proposed alignment of BOT [25] and SAREF4Building [24].

Subject	**Predicate**	**Object**
Class level:		
s4bldg:Building	rdfs:subClassOf	bot:Building
s4bldg:PhysicalObject	rdfs:subClassOf	bot:Element
s4bldg:Sensor	rdfs:subClassOf	bot:Element
s4bldg:Actuator	rdfs:subClassOf	bot:Element
s4bldg:BuildingSpace	rdfs:subClassOf	bot:Space
Object property level:		
s4bldg:hasSpace	rdfs:subPropertyOf	bot:containsZone
s4bldg:contains	rdfs:subPropertyOf	bot:containsElement

4.2 BRICK Uniform Schema for Representing Metadata in Buildings

The BRICK ontology [3,4] is an ontology, which focuses on the description of building management systems and their domain concepts such as data points, HVAC equipment and the topology of the building. The defined and revised alignments are reported in Table 3 and explained in the following.

The alignments are defined by using subsumption on class and object property level. A `brick:Location` is considered as a specialisation of `bot:Zone` as it is used as a super-concept from other concepts describing topological aspects of a building. Hence, the alignment of `brick:Building`, `brick:Floor`, `brick:Basement`, `brick:Outside`, `brick:Room`, `brick:Space` and `brick:Wing` is straightforward. A `brick:Zone` is considered to be a specialisation of a `bot:-Space` as BRICK uses this concepts to refer to HVAC zones often used in the context of the control of a building. The `brick:Equipment` of a building and `brick:Points` are considered as specialisations of `bot:Element`. In particular this holds for the concept `brick:Point` as BRICK describes for instance sensors or meters as tangible objects, which are located in some zone or space.

The semantics of `brick:contains` can be directly mapped to `bot:contains-Element` and, hence, a specialisation is defined. Interesting is the `brick:has-Part` object property defined in BRICK. The property can be used to relate `brick:Equipment` to `brick:Sensors`, `brick:Equipment` to `brick:Equipment` or `brick:Locations` to `brick:Locations` [3], hence, it qualifies for an extension of `bot:containsElement`, `bot:containsZone` and `bot:hasSubElement` as defined. Similar semantics apply for the `brick:hasPoint` object property, which can be specialised from `bot:containsZone` and `bot:containsElement` as it allows to relate `brick:Equipment` to `brick:Sensors` and `brick:Locations` to `brick:-Sensors`.

Table 3. Proposed alignment of BOT [25] and BRICK [3,4].

Subject	**Predicate**	**Object**
Class level:		
brick:Location	rdfs:subClassOf	bot:Zone
brick:Building	rdfs:subClassOf	bot:Building
brick:Floor	rdfs:subClassOf	bot:Storey
brick:Basement	rdfs:subClassOf	bot:Space
brick:Outside	rdfs:subClassOf	bot:Space
brick:Room	rdfs:subClassOf	bot:Space
brick:Space	rdfs:subClassOf	bot:Space
brick:Wing	rdfs:subClassOf	bot:Space
brick:Zone	rdfs:subClassOf	bot:Space
brick:Equipment	rdfs:subClassOf	bot:Element
brick:Point	rdfs:subClassOf	bot:Element
Object property level:		
brick:contains	rdfs:subPropertyOf	bot:containsElement
brick:hasPart	rdfs:subPropertyOf	bot:containsZone
brick:hasPart	rdfs:subPropertyOf	bot:containsElement
brick:hasPart	rdfs:subPropertyOf	bot:hasSubElement
brick:hasPoint	rdfs:subPropertyOf	bot:containsZone
brick:hasPoint	rdfs:subPropertyOf	bot:containsElement

4.3 DogOnt - Ontology Modeling for Intelligent Domotic Environments

DogOnt ontology is an ontology to formally describe the domain of domotic devices in home appliances. It is initially described in Bonino & Corno [6] but has since undergone many revision and extensions. The most recent version as of writing (4.0.1) can be obtained from a remote repository[5].

A number of concepts of the ontology can be specialised from BOT. The defined alignments are reported in Table 4. In particular the ontology describes the concepts `dogont:Building`, `dogont:Storey`, `dogont:Room`, which can be mapped to respective BOT concepts. The general concept of `dogont:Environment` can be seen of a generalisation of the `bot:Zone` concept as defined. Interesting is the definition of the concepts `dogont:Ceiling` and `dogont:Floor` as areas bounding a room. This complies to the definition of `bot:Interface` and can be aligned by specialisation. To reflect the different semantics the different sub-concepts of `dogont:UnControllable` need to be separately specialised (e.g. `dogont:Furniture`).

In terms of aligning object properties a number of specialisation can be found. The object property `dogont:contains` is used in DogOnt to describe that some tangible object is fully contained in a `dogont:BuildingEnviroment`. Essentially this is the semantics of `bot:containsElement`. The object properties `dogont:belongsTo` and `dogont:hasWallOpening` allow to describe composition of classes which specialised from `bot:Element`. Hence, they are specialised from `bot:hasSubElement`, potentially its inverse where needed. Interesting are also the `dogont:floorOf` and `dogont:ceilingOf` object properties, which qualify as specialisation of `bot:interfaceOf` together with the specialisation of `dogont:Ceiling` and `dogont:Floor` as `bot:Interface` as defined above.

4.4 ThinkHome Ontology

The ThinkHome ontologies [27] are a family of ontologies to describe smart home systems formally and link this with adjacent domains. A detailed description of the ontologies can be found in its documentation and reviews [27,22,28]. Alignments are defined to the BuildingOntology of the family of ontologies, which has been derived from the gbXML format[6].

The common concepts of `th:Campus`, `th:Building`, `th:BuildingStorey`, `th:Opening`, `th:Space`, `th:Zone` and `th:Equipment` can be specialised directly from BOT. Interesting are the concepts `th:Construction`, `th:Layer`, `th:Material`, which refer to different layers of a wall needed for instance in building performance simulation. The semantics comply to `bot:Interface` and hence a specialisation is defined.

A number of object properties are defined in ThinkHome, which describe the containment of an element in a zone, a zone in a zone or composition of

[5] `https://github.com/iot-ontologies/dogont`, Last accessed: 20 May 2019

[6] `http://www.gbxml.org/`, Last accessed: 20 May 2019

Table 4. Proposed alignment of BOT [25] and DogOnt [6].

Subject	**Predicate**	**Object**
Class level:		
dog:Building	rdfs:subClassOf	bot:Building
dog:Storey	rdfs:subClassOf	bot:Storey
dog:Room	rdfs:subClassOf	bot:Space
bot:Zone	rdfs:subClassOf	dog:Environment
dog:BuildingEnvironment	rdfs:subClassOf	bot:Zone
dog:Room	rdfs:subClassOf	bot:Space
dog:Balcony	rdfs:subClassOf	bot:Zone
dog:Terrace	rdfs:subClassOf	bot:Zone
dog:Controllable	rdfs:subClassOf	bot:Element
dog:Device	rdfs:subClassOf	bot:Element
dog:TechnicalSystem	rdfs:subClassOf	bot:Element
dog:Vertical	rdfs:subClassOf	bot:Element
dog:Furniture	rdfs:subClassOf	bot:Element
dog:Ceiling	rdfs:subClassOf	bot:Interface
dog:Floor	rdfs:subClassOf	bot:Interface
Object property level:		
dog:contains	rdfs:subPropertyOf	bot:containsElement
dog:belongsTo	rdfs:subPropertyOf	owl:inverseOf bot:hasSubElement
dog:floorOf	rdfs:subPropertyOf	bot:interfaceOf
dog:ceilingOf	rdfs:subPropertyOf	bot:interfaceOf
dog:hasWall	rdfs:subPropertyOf	bot:adjacentElement
dog:hasWallOpening	rdfs:subPropertyOf	bot:hasSubElement

elements. Hence, the object properties are specialised from the respective object properties in BOT as reported in Table 5.

Table 5. Proposed alignment of BOT [25] and ThinkHome [27].

Subject	**Predicate**	**Object**
Class level:		
th:Campus	rdfs:subClassOf	bot:Site
th:Building	rdfs:subClassOf	bot:Building
th:BuildingStorey	rdfs:subClassOf	bot:Storey
th:Opening	rdfs:subClassOf	bot:Element
th:Space	rdfs:subClassOf	bot:Space
th:Zone	rdfs:subClassOf	bot:Zone
th:Construction	rdfs:subClassOf	bot:Interface
th:Layer	rdfs:subClassOf	bot:Interface
th:Material	rdfs:subClassOf	bot:Interface
th:SpaceBoundary	rdfs:subClassOf	bot:Interface
th:Surface	rdfs:subClassOf	bot:Interface
th:Equipment	rdfs:subClassOf	bot:Element
Object property level:		
th:containsBuilding	rdfs:subPropertyOf	bot:hasBuilding
th:containsBuildingStorey	rdfs:subPropertyOf	bot:hasStorey
th:containsSpace	rdfs:subPropertyOf	bot:hasSpace
th:containsSpaceBoundary	rdfs:subPropertyOf	bot:interfaceOf
th:containsLighting	rdfs:subPropertyOf	bot:containsElement
th:containsHydronicLoopEquipment	rdfs:subPropertyOf	bot:hasSubElement
th:containsAirLoopEquipment	rdfs:subPropertyOf	bot:hasSubElement
th:hasDefinedAdjacentSpace	rdfs:subPropertyOf	owl:inverseOf bot:adjacentZone
th:isDefinedAdjacentSurfaceOf	rdfs:subPropertyOf	bot:interfaceOf
thsv:isEquipmentOf	rdfs:subPropertyOf	owl:inverseOf bot:hasElement

4.5 Industry Foundation Classes 4 Addendum 2

No new or revised alignments to the OWL version of the Industry Foundation Classes (IFC) [17,22] from BOT are found in this work in comparison to the initial mapping [29]. However, the alignments are changed to subsumption as mentioned above.

4.6 DERI Room Ontology

The DERI Room ontology [8] has been added to this study as is represents a light weight vocabulary, compared to the other ontologies, dedicated to the description of buildings. The found alignments are documented in Table 7 and almost all classes and object properties could be specialised from BOT.

Table 6. Proposed alignment of BOT [25] and ifcOWL4_Add2 [22].

Subject	Predicate	Object
ifc:IfcSite	rdfs:subClassOf	bot:Site
ifc:IfcBuilding	rdfs:subClassOf	bot:Building
ifc:IfcBuildingStorey	rdfs:subClassOf	bot:Storey
ifc:IfcSpace	rdfs:subClassOf	bot:Space
ifc:IfcElement	rdfs:subClassOf	bot:Element

Table 7. Proposed alignment of BOT [25] and DERI Room [8].

Subject	Predicate	Object
Class level:		
rooms:Site	rdfs:subClassOf	bot:Site
rooms:Building	rdfs:subClassOf	bot:Building
rooms:Floor	rdfs:subClassOf	bot:Storey
rooms:FloorSection	rdfs:subClassOf	bot:Storey
rooms:Desk	rdfs:subClassOf	bot:Element
rooms:Room	rdfs:subClassOf	bot:Space
Object property level:		
rooms:contains	rdfs:subPropertyOf	bot:containsZone

4.7 Summary

The results of the manually defined and revised alignments are summarised in Table 8. The respective domain ontologies are denoted and the considered version of the ontology, if applicable. All defined mappings are defined in a separated ontology file and the respective ontology is checked for consistency by invoking a OWL DL reasoner (Pellet [31]) and looking for inconsistencies. The total number of alignments between concepts and object properties are reported. It should be noted that the total number does not qualify as a metric to determine if BOT can be extended very well to the respective domain. In comparison to the last study [29] it is interesting that the number of ontologies, where a mapping on the object property level is possible has been significantly increased.

5 Automated Alignment

As described in Section 3 a study is conducted in this work using the tool AgreementMakerLight [13] to automatically derive ontology alignments. Figure 1 shows as an example the result reported by the tool for matching BOT and the ThinkHome ontology.

The tool reports the found alignments in the graphical user interface as well as exports them in RDF format. All automatically found alignments are reported in Table 9. The automated ontology matching tool has found from zero up to three alignments between concepts of the respective ontologies. No alignments

Table 8. Evaluation of manually defined alignments to BOT version v0.3.0. Consistency - Invoking Pellet [31] reasoner on the respective alignment ontology including `owl:imports` did not return any faults.

Domain Ontology	**Version**	**Consistency**	**No. of Alignments** `owl:Class`	**No. of Alignments** `owl:objectProperty`
S4BLDG [24]	1	OK	5	2
BRICK [3,4]	1.0.2	OK	11	6
DogOnt [6]	4.0.1	OK	15	6
ThinkHome [27]	1.12	OK	12	10
ifcOWL [22]	4Add2	OK	5	0
DERIRoom [8]	-	OK	6	1

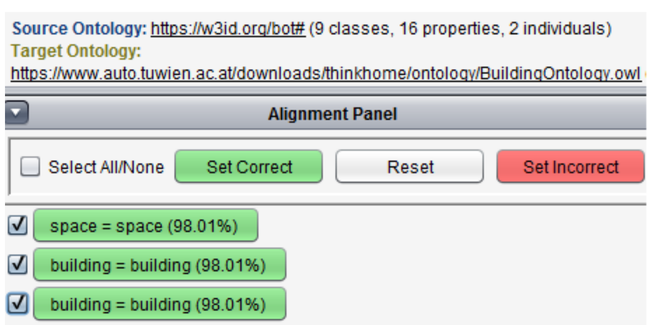

Fig. 1. Results reported for automatically matching BOT and ThinkHome ontologies by the AgreementMakerLight tool [13].

between object properties are found. The reported suggested alignment is always equivalence. One false alignment is reported mapping a `bot:Space` to `ifc:IfcSpaceType`.

AgreementMakerLight implements three types of primary matching algorithms: a lexical matcher, mediating matcher and word matcher and one secondary type matching algorithm: a parametric string matcher [13]. An in depth treatment of the matching methods is beyond the scope of this paper. All algorithms take as an input the two to-be-aligned ontologies. The primary matching algorithms compare obtained terms and assert alignments if a similarity measure exceeds a threshold with different complexities. Hence, it should be noted that the chosen parameterisation of the thresholds, etc. has an impact on the results and should be studied in more detail on a elaborated benchmark defined for the AEC/FM domain. In the initial experiments conducted in this study the default suggested values are utilised.

Table 9. Results reported for automatically matching BOT to the respective listed domain ontology. (1) - Domain Ontology, (2) - Number of found alignments, X - Falsely reported alignment.

(1)	**(2)**	**BOT Concept**	**Target concept**	**Type**
S4BLDG [24]	1	`bot:Building`	`s4bldg:Building`	Equivalence
BRICKFrame [3,4]	-	-	-	-
DogOnt [6]	2	`bot:Storey`	`dog:Storey`	Equivalence
		`bot:Building`	`dog:Building`	Equivalence
ThinkHome [27]	3	`bot:Space`	`th:Space`	Equivalence
		`bot:Building`	`th:Building`	Equivalence
		`bot:Building`	`thsv:Building`	Equivalence
ifcOWL [22]	2	`bot:Element`	`ifc:IfcBuildingElement`	Equivalence
	X	`bot:Space`	`ifc:IfcSpaceType`	Equivalence
DERIRoom [8]	2	`bot:Site`	`rooms:Site`	Equivalence
		`bot:Building`	`rooms:Building`	Equivalence

6 Discussion

Automated ontology matching is a well-known discipline and the topic is researched since decades. A plethora of tools is available, see e.g. [1]. In this study only one tool has been used. A detailed study of methods and tools should be conducted to further clarify the abilities of automated ontology matching methods for their application in the AEC/FM domain.

The ontologies considered in this study mainly reside from the building automation domain. This is mainly motivated by the authors expertise and research interest. However, other AEC/FM domains should be included in future studies on automated ontology matching.

7 Conclusion

Within this paper manually defined and automatically obtained alignments between domain ontologies from the Architecture, Engineering, Construction/ Facility Management (AEC/FM) domain to the Building Topology Ontology (BOT) [25] are compared. The manual definition of alignments between ontologies is a tedious task and almost as difficult as developing ontologies from scratch. Initial experiments show that automated matching methods can support finding alignments. The results are promising to also support not only the alignment of domain ontologies but the revision of alignments, e.g. because of schema level updates.

The presented study can only be seen as a starting point and the following open questions for future research remain:

- There is a strong need for the definition of a well-defined benchmark from AEC/FM domain, potentially including building product data, to establish the attention of ontology matching experts;
- Addittional sub-domains of the AEC/FM domain should be added in future studies;
- As ontologies evolve over time a future question is, if existing alignments can be reused as a starting point ("hot-start") for matching methods.

Acknowledgements

This paper documents work conducted in a collaborative effort by the W3C LBD CG. The author gratefully acknowledges financial support from MOEEBIUS project, a Horizon 2020 research and innovation program under grant agreement No. 680517 and the initiative Mittelstand 4.0 by the German Federal Ministry for Economic Affairs and Energy.

References

1. Algergawy, A., Cheatham, M., Faria, D., Ferrara, A., Fundulaki, I., et al.: Results of the ontology alignment evaluation initiative 2018. In: Proc. of OM. pp. 1–41. Monterey, USA (2018)
2. Anumba, C.J., Issa, R.R., Pan, J., Mutis, I.: Ontology based information and knowledge management in construction. Construction Innovation **8**(3), 218–239 (2008). https://doi.org/10.1108/14714170810888976
3. Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Berges, M., Culler, D., Gupta, R., Kjærgaard, M.B., Srivastava, M., Whitehouse, K.: Brick: Towards a unified metadata schema for buildings. In: Proc. of BuildSys. Palo Alto, USA (2016). https://doi.org/10.1145/2993422.2993577
4. Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Berges, M., Culler, D., Gupta, R.K., Kjærgaard, M.B., Srivastava, M., Whitehouse, K.: Brick : Metadata schema for portable smart building applications. Applied Energy **226**, 1273–1292 (2018). https://doi.org/10.1016/j.apenergy.2018.02.091

5. Bellini, P., Benigni, M., Billero, R., Nesi, P., Rauch, N.: Km4city ontology building vs data harvesting and cleaning for smart-city services. Journal of Visual Languages & Computing **25**(6), 827–839 (2014). https://doi.org/10.1016/j.jvlc.2014.10.023
6. Bonino, D., Corno, F.: DogOnt - Ontology Modeling for Intelligent Domotic Environments. Lecture Notes in Computer Science **5318**, 790–803 (2008). https://doi.org/10.1007/978-3-540-88564-1_51
7. Costin, A., Eastman, C.: Need for interoperability to enable seamless information exchanges in smart and sustainable urban systems. Journal of Computing in Civil Engineering **33**(3), 1–14 (2019)
8. Cyganiak, R.: Buildings and rooms vocabulary. `http://vocab.deri.ie/rooms`, Last accessed: 20 May 2019 (2012), Digital Enterprise Research Institute (DERI), Galway, Ireland
9. Daniele, L., den Hartog, F., Roes, J.: Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology. In: Cuel, R., Young, R. (eds.) Proc. of FOMI. pp. 100–112. Springer International Publishing, Cham, Switzerland (August 5 2015). https://doi.org/10.1007/978-3-319-21545-7_9
10. Domingue, J., Fensel, D., Hendler, J.A.: Handbook of semantic web technologies. Springer, Berlin, Germany (2011). https://doi.org/10.1007/978-3-540-92913-0
11. Espinoza-Arias, P., Poveda-Villalon, M., Garcia-Castro, R., Corcho, O.: Ontological representation of smart city data: From devices to cities. Applied Sciences **9**(1) (2018). https://doi.org/10.3390/app9010032
12. Euzenat, J., Shvaiko, P.: Ontology matching, vol. 18. Springer, Heidelberg, Germany, 2nd edn. (2013). https://doi.org/10.1007/978-3-642-38721-0
13. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The AgreementMakerLight Ontology Matching System. In: Proc. of ODBASE. Springer, Berlin, Germany (2013). https://doi.org/10.1007/978-3-642-41030-$7_3$8
14. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. In: Proc. of AAAI. pp. 33–40. Stanford, USA (March 24-26 1997)
15. Gupta, S., Szekely, P., Knoblock, C.A., Goel, A., Taheriyan, M., Muslea, M.: Karma: A system for mapping structured sources into the semantic web. In: Proc. of ESWC. pp. 430–434 (2015)
16. Gyrard, A., Zimmermann, A., Sheth, A.: Building iot-based applications for smart cities: How can ontology catalogs help? IEEE Internet of Things Journal **5**(5), 3978–3990 (Oct 2018). https://doi.org/10.1109/JIOT.2018.2854278
17. ISO: ISO 16739 - Industry Foundation Classes (2013)
18. Musen, M.A., Team, T.P.: Protégé Ontology Editor. In: Dubitzky, W., Wolkenhauer, O., Cho, K.H., Yokota, H. (eds.) Encyclopedia of Systems Biology. pp. 1763–1765. Springer, New York, USA (2013). https://doi.org/10.1007/978-1-4419-9863-7_1104
19. Noy, N.F., McGuinness, D.L.: Ontology Development 101: A Guide to Creating Your First Ontology. Tech. Rep. SMI-2001-0880, Stanford University, Stanford, USA (2001)
20. Otero-Cerdeira, L., Rodríguez-Martínez, F., Gómez-Rodríguez, A.: Definition of an ontology matching algorithm for context integration in smart cities. Sensors **14**(12), 23581–23619 (2014)
21. Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Valencia-Requejo, T., Gómez-Rodríguez, A.: A new similarity measure for an ontology matching system. In: Fred, A., Dietz, J.L.G., Aveiro, D., Liu, K., Filipe, J. (eds.) Knowledge Discovery, Knowledge Engineering and Knowledge Management. pp. 257–272. Springer International Publishing, Cham, Switzerland (2015)

22. Pauwels, P., Terkaj, W.: EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. Automation in Construction **63**, 100–133 (2016). https://doi.org/10.1016/j.autcon.2015.12.003
23. Pauwels, P., Zhang, S., Lee, Y.C.: Semantic web technologies in AEC industry: A literature overview. Automation in Construction **73**, 145–165 (2017). https://doi.org/10.1016/j.autcon.2016.10.003
24. Poveda Villalon, M., Garcia Castro, R.: SAREF extension for building devices (2017), `http://ontoology.linkeddata.es/publish/saref4bldg/index-en.html`, [Online; last accessed 2017-09-12]
25. Rasmussen, M.H., Pauwels, P., Hviid, C.A., Karlshøj, J.: Proposing a Central AEC Ontology That Allows for Domain Specific Extensions. In: Proc. of LC3. vol. 1, pp. 237–244. Heraklion, Greece (2017). https://doi.org/10.24928/JC3-2017/0153
26. Rasmussen, M.H., Pauwels, P., Lefrançois, M., Schneider, G.F., Hviid, C.A., Karlshøj, J.: Recent changes in the Building Topology Ontology. In: Proc. of LDAC. Dijon, France (November 2017)
27. Reinisch, C., Kofler, M.J., Iglesias, F., Kastner, W.: Thinkhome energy efficiency in future smart homes. EURASIP Journal on Embedded Systems **2011**(1), 104617 (2011). https://doi.org/10.1155/2011/104617
28. Schneider, G.F., Pauwels, P., Steiger, S.: Ontology-based Modeling of Control Logic in Building Automation Systems. IEEE Transactions on Industrial Informatics **13**(6), 3350–3360 (2017). https://doi.org/10.1109/TII.2017.2743221
29. Schneider, G.F.: Towards Aligning Domain Ontologies with the Building Topology Ontology. In: Proceedings of the 5th Linked Data in Architecture and Construction Workshop (LDAC). pp. 1–8. Dijon, France (2017). https://doi.org/10.13140/RG.2.2.21802.52169
30. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Transactions on knowledge and data engineering **25**(1), 158–176 (2013). https://doi.org/10.1109/TKDE.2011.253
31. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-Dl reasoner. Web Semantics: science, services and agents on the World Wide Web **5**(2), 51–53 (2007)
32. W3C LBD CG: Building Data on the Web Working Group Charter. `https://w3c-lbd-cg.github.io/lbd/charter/`, Last accessed: 20 May 2019 (2017), last accessed: 11 July 2017