Onto4Reuse: Towards an Ontology Reuse Framework for Knowledge-intensive Software Engineering

João Moreira[1,*,†], Alex Donkers[2,†], Pieter Pauwels[2], Esra Bektas[3] and Tim van Ee[3]

[1]*Semantics, Cybersecurity and Services (SCS), University of Twente, Drienerlolaan 5, Enschede, 7522 NB, The Netherlands*
[2]*Information Systems in the Built Environment, Eindhoven University of Technology, Groene Loper 6, 5600MB Eindhoven, The Netherlands*
[3]*TNO - Digital Built Environment, Molengraaffsingel 8, 2629 JD, Delft, The Netherlands*

Abstract

Ontologies are a primary resource in knowledge-intensive software. However, reusing ontologies is challenging, which leads to redundant efforts, wasted resources, and hindered technological progress. This issue becomes even more severe and evident in the system engineering field, once engineering tasks rely on precise and well-structured data models and knowledge representation techniques. Particularly, the construction and infrastructure domain heavily relies on these structured data and their associated models to annotate and classify the assets, and these data models are increasingly being represented using domain ontologies. This paper discusses a research agenda to address the general problem of ontology reuse, emphasizing to the system engineering areas of infrastructure asset management and modular building construction. We introduce the Onto4Reuse framework that aims at supporting software engineers in designing and embedding operational ontologies in software systems by leveraging on existing ontologies. The Onto4Reuse framework covers good practices on how an ontology should be developed in a way that it can be reused with lower efforts in other contexts. To achieve this goal, the Onto4Reuse framework is leveraged by four pillars: (1) Ontology design patterns; (2) ontology matching based on foundational ontology; (3) reverse engineering for grounding operational ontologies in a foundational ontology; and (4) the application of the Findable, Accessible, Interoperable and Reusable (FAIR) data principles. These techniques are the foundations for developing ontology-driven software systems, and some functions will be exposed as services, which will be then applied to the aforementioned engineering cases in real-life utilization settings to validate the framework.

Keywords

Ontology Reuse, Unified Foundational Ontology, Ontology Design Patterns, Ontology Matching

1. Introduction

Engineering tasks rely on well-structured data models and knowledge representation techniques. Among others, the construction and infrastructure domain heavily relies on these structured data, to annotate and classify all the assets in their domain. These data models are represented using ontologies, which can be understood as formally structured vocabularies of a certain

FOMI 2024: 13th International Workshop on Formal Ontologies Meet Industry
*Corresponding author.
†These authors contributed equally.
j.luizrebelomoreira@utwente.nl (J. Moreira); a.j.a.donkers@tue.nl (A. Donkers); p.pauwels@tue.nl (P. Pauwels); esra.bektas@tno.nl (E. Bektas); tim.vanee@tno.nl (T. v. Ee)
0000-0002-4547-7000 (J. Moreira); 0000-0002-8809-3277 (A. Donkers); 0000-0001-8020-4609 (P. Pauwels); 0000-0001-8020-4609 (E. Bektas); 0000-0001-8020-4609 (T. v. Ee)

domain model (e.g. road infrastructure or modular housing). Building such ontologies is a very expensive process, and therefore it is recommended to reuse these ontologies as best as possible, thus minimizing redundancy and maximizing efficiency in knowledge representation systems. To enable this Ontology Reuse (OR), an ontology usually follows a recurring Ontology Design Pattern (ODP) [1] and can be matched with others through ontology alignments based on foundational categories [2], to then facilitate the sharing and integration of domain knowledge across different applications and domains, such as knowledge graphs and digital twins [3]. This fosters interoperability and collaboration in various knowledge-intensive engineering domains, such as infrastructure condition assessment and modular construction engineering.

Several ontologies have been developed over the years in the domain of civil infrastructure and construction engineering but reusing and matching these ontologies based on current methodologies is limited as these methodologies focus primarily on similarity in the semantics of the terms. The term "building" is described in at least six existing ontologies. Yet, this does not mean that these terms represent the same concept or that these concepts share the same statements and can therefore be simply matched as equivalent classes (e.g. using *owl:equivalentClass*). Matching classes as equivalent merely based on shared semantics may cause false logic. Variations in structure, naming conventions, and modeling choices further complicate OR. The consequences of ineffective reuse are substantial, resulting in redundant efforts, wasted resources, and hindered technological progress. Different motivations and implementations have led to diverse OR approaches, including both the direct reuse of standard ontologies as well as the use of tailored ontology designs that are aligned with standards, or hybrid approaches [4, 5, 6]. This often still leads to the creation of slightly different ontologies. Furthermore, it is often challenging (1) to find the ontologies that one would want to reuse, despite available platforms like Linked Open Vocabularies (LOV) [1], and (2) to evaluate the quality or ease of reuse for available ontologies.

This paper introduces an implementation agenda to develop the first ontology engineering (OE) framework, coined as Onto4Reuse, that will emphasize on how software engineers can design and embed operational ontologies in software systems by leveraging existing ontologies, allowing future research to use a reliable and realistic reference for OR. Onto4Reuse framework can support organizations to embed existing ontologies into their Software Engineering (SE) lifecycle. Leveraging such a framework, end users will reduce the costs of software development significantly in the case of knowledge-intensive projects, particularly engineering projects.

Onto4Reuse is grounded in four pillars: (1) application of ontology design patterns [1]; (2) ontology matching based on a foundational ontology [2]; (3) semi-automatic reverse engineering to infer ontological categories of existing operational ontologies in OWL [7]; and (4) the application of the FAIR data principles [8, 9]. This approach will be tested and validated in the context of the construction industry, with a case in infrastructure asset management and in modular building construction, which are both areas that are known to experience many problems related to repeatedly re-engineering structured domain knowledge [10, 11].

This paper is structured as follows. Section 2 motivates the research on OR in software development and covers the background. Section 3 describes the problem and explains how the four pillars can improve ontology design for higher reusability. Section 4 presents the

[1] https://lov.linkeddata.es/dataset/lov/

implementation agenda through the development of the Onto4Reuse framework, and Section 5 discusses how the framework services support two relevant application cases. Section 6 concludes this paper.

2. Reuse in Ontology Engineering

Ontologies are at the core of software development, as they provide formally defined data structures that can be used reliably by those software. However, defining and using these ontologies is challenging. One key first step is *applying* an ontology. This involves the use of an existing ontology within a specific context or application, typically to provide structure and semantic meaning to data or knowledge within that context. This is strongly supported and advocated in many domains, including the construction industry, through the use of a standard ontology, such as the Industry Foundation Classes (IFC), for data exchange processes. It is strongly recommended to rely on existing ontologies and apply them to future cases [10].

However, in a large majority of cases, the ontology that is meant to be applied does not fully match the considered case, and an extension, modification, or just a subset is needed. In such case, a number of approaches can be followed: 1) the software engineer moves away from OR and starts developing an entirely new ontology (the ”11th standard”); 2) the software engineer makes minor local changes to the ontology, and potentially notifies the creator of the original ontology; and 3) the software engineer combines the ontology with one or more other ontologies that make up for missing or to be modified parts, such as in [8]. This can be recognized in efforts where a standard ontology is adopted, e.g. IFC, yet extended with a few more specific properties or classes for object representation. *Reusing* an ontology entails adopting an existing ontology in the development of new systems, often involving integration with other ontologies or customization to fit the requirements of the new context, while striving to maintain compatibility and interoperability with the original ontology. Such an example can be found in the attempt to reuse the IFC ontology in the creation of the SAREF4BLDG ontology[2], where a subset of classes and properties were kept, while referring to the source ontology.

OR is an important topic within the field of OE, and, even though considerable research has been done already on how to reuse existing ontologies [4, 5, 12, 6], this still represents a significant challenge, especially also in practice. First, ontologies are often domain-specific, and developed with a specific purpose to capture knowledge within a particular context or industry [2]. This specificity means that reusing an ontology from one domain in another always requires (extensive) modifications, often rendering the reuse effort inefficient and time-consuming, since at best only a few abstract ODPs can be reused, if they are even recognized. Second, semantics are contextual and ontologies are typically created by different individuals or groups with varying points of view to fit specific purposes - so capturing different sets of properties and focusing on different aspects of the domain - along with different design principles and methodologies, leading to disparities in structure, naming conventions, and modeling choices [6]. This can be recognized in the overlaps and differences for a building structure in the BOT, SAREF, IFC, and BRICK ontologies [13]. These inconsistencies make seamless integration and reuse across ontologies a complex task. At best, only basic class

[2] https://saref.etsi.org/saref4bldg/v1.1.2/

equivalence alignments can be made, which represent often less than 10% of the ontology scope. Third, evolving technologies and changing domain-specific requirements can quickly render existing ontologies obsolete, requiring continuous updates and modifications, further complicating ontology management and governance (see for example the IFC ontology versions). Finally, in some specific cases, ontologies cover information that are sensitive to companies and, therefore, are developed internally, e.g., for improved control and security, which impacts OR.

3. Problem Definition

The societal and economic consequences of not effectively reusing ontologies are substantial. Inefficient OR results in redundant efforts, as ontology engineers are forced to reinvent the wheel knowingly, thus spending valuable time and resources that could be directed to value creation, be it societal or economic. At the moment, most infrastructure management organizations have built their own ontologies, in parallel to each other; furthermore, several contractor and engineering firms, as well as city councils are building their own ontologies. This leads to a significant waste of intellectual capital and financial investments, hindering technological progress. Consequences of poor OR across multiple domains (1) hampers the retrieval of ontology design patterns that have proven value in knowledge representation, and (2) hampers collaboration between different sectors as there exist little similarity matching mechanisms beyond string-based approaches [5]. Ultimately, the failure to address the challenges of OR not only impacts technological advancements, but also has far-reaching societal and economic repercussions by inhibiting efficient knowledge sharing and stifling innovation in a rapidly evolving digital landscape [4].

In the near future, ontology alignment and reuse will become also more important in the buildingSMART Open BIM standardization community. Their Industry Foundation Classes (IFC) Schema, now in version 4.3 covering buildings and civil infrastructures, can be regarded as a predefined ontology for the AEC/FM sector[3]. The buildingSMART Data Dictionary (bSDD) is a key mechanism to define extensions - by project, organization, country, i.e. your own ontology - to IFC. Ontology alignment will be relevant not only to connect such extensions in the right way to IFC but especially also to interrelate those extensions promoting the reuse of each other. Typically the bSDD dictionaries are generated ('converted') from their own 'bSDD Excel Template' or via specialized transformers like for CEN TC442 WG4 Semantic Modeling and Linking (SML)-standard compliant ontologies[4]. The European SML itself is covered in the Netherlands by the NEN-2660-2 standard applied in all major ontology/object type library (OTL) initiatives in Dutch industry (RWS-OTL[5], IMBOR[6], Waternet-OTL[7], Amsterdam-OTL[8], TenneT-OTL [14], GWSW ontology[9], etc.).

Developing ontology *for* reuse is a driver that involves designing a modular, well-structured

[3]https://ifc43-docs.standards.buildingsmart.org/
[4]https://technical.buildingsmart.org/services/bsdd/data-structure/
[5]https://otl.rws.nl/
[6]https://www.crow.nl/thema-s/management-openbare-ruimte/imbor/actuele-versie-imbor
[7]https://otl.waternet.nl/
[8]https://amsterdam.otl-viewer.com/
[9]https://data.gwsw.nl/

ontology that is adaptable to various domains and applications, aiming to maximize its utility across different contexts while minimizing redundancy. It requires careful consideration of common concepts, relations, and best practices to ensure interoperability and facilitate seamless integration into diverse knowledge-based systems [15]. In the last years, a number of standardization organizations have been working on standardizing ontologies, which in theory should be ontologies designed for reuse. Standardization efforts play a pivotal role in enhancing the interoperability and adoption of reusable ontologies by establishing consistent frameworks and vocabularies for knowledge representation and exchange, and therefore, with the driver on ontology for reuse. For example, the European Telecommunication Standardization Institute (ETSI) produced the Smart Applications REFerence (SAREF) ontology, while the Health Level Seven International (HL7) produced the Fast Healthcare Interoperability Resources (FHIR), besides the several W3C recommendations, such as PROV (for data provenance) and DCAT (for data catalogs), among others [3]. However, it is still a challenge to apply, use, and reuse such standardized ontologies. Therefore, we frame our problem through this main research question: *how to improve the reuse of ontologies for embedding in software systems through OE good practices for reuse?*

4. Onto4Reuse: Reuse Ontologies in Software Engineering

4.1. Framework Requirements

The Onto4Reuse framework aims at supporting software engineers to efficiently engineer ontologies for reuse, and effectively embed operational ontologies during software development. The Onto4Reuse framework is guided by the reference architecture illustrated in Figure 1. The framework should provide (R1) clarified guidelines on OR methods, (R2) OR quality metrics and a measuring approach, (R3) a set of software tools as services to support the software engineer in OR practice through Model-Driven Software Engineering (MDSE). These services should be validated in the context of OR in infrastructure condition assessment and modular construction engineering.

The first element of Onto4Reuse reference architecture on OR Theory and Framework Requirements aims at investigating the OR problem, covering good practices and required functionalities of the framework, addressing R1 requirement. The second element on OPL with Ontology Matching aims at specifying ontology patterns and matching approach, addressing R2 requirement. The third element on OR services as MDSE aims at addressing R3 requirement. The fourth and fifth elements cover the utilization of the framework in the two engineering cases.

Onto4Reuse framework targets the three key parts that are most relevant for OR: ontology selection, ontology integration, and ontology access. Available OR methods are evaluated and refined throughout those three phases. Recommendations for OR include task-based ontology selection through effective competency questions; trustworthiness of ontology access solutions through engineering and social negotiation; usability of ontology integration results (e.g., documentation) and querying data using integrated ontologies; and knowledge stability to implement OR through raised awareness of cognitive analysis in ontology design for semantic interoperability [4].

We intend to implement the Onto4Reuse framework on top of an existing OE methodology,

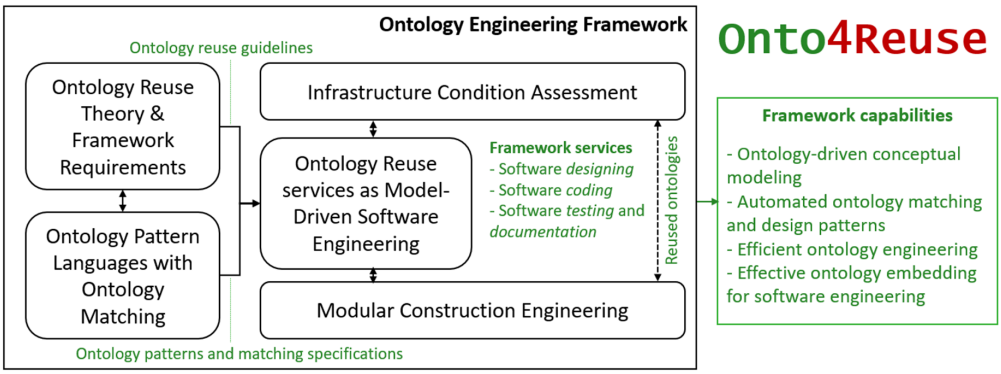

Figure 1: Onto4Reuse reference architecture for effective ontology reuse in software engineering

namely the Systematic Approach for Building Ontologies (SABiO) [16]. We chose SABiO because it covers the most common activities in ontology engineering - ontology requirements, capture/formalization, design, implement and test. SABiO is well aligned to SE methodologies and can be easily translated to other ontology engineering methodologies [17]. In addition, we have been applying SABiO for a decade now, which give us a solid methodological basis. We will emphasize supporting OR by leveraging our expertise with the UFO and the ontology-driven conceptual modeling practice based on UFO's ontological language (OntoUML) [15]. One of the assumptions is that enriching domain ontologies with foundational ones, such as UFO, can promote their use as semantic bridges in matching domain ontologies [2].

We plan to integrate the results of previous research in four pillars as illustrated in Figure 2: (1) implement Ontology Pattern Language (OPL) for ontology design patterns [1]; (2) improve ontology matching approaches by leveraging on UFO [2]; (3) reuse of existing operational ontology formalized in OWL through a semi-automatic reverse engineering approach that infers ontological categories of OWL classes using foundational rules, coined as Scior, of the lightweight UFO (gUFO) [7]; and (4) the application of the FAIR data principles in the ontologies catalog [8, 9], with emphasis on the three key parts for OR.

Through Onto4Reuse we propose to apply the four aforementioned research pillars through a holistic approach based on the ontology-driven conceptual modeling practice with ontology patterns (at design-time), and semi-automatic ontology matching based on reverse engineering (at runtime). This will guide the software engineer to reuse ontologies in specific contexts, supporting understanding of the role of ontologies for their specific purposes. Our outcomes will lay the groundwork for an OR system based on service-oriented architecture principles and implemented through MDSE transformations [18], which organizations can embed into their software development lifecycle to obtain a comprehensive view of ontologies related to their project needs and running systems. By leveraging this system, end users will largely reduce the costs of developing ontology-based software while minimizing the risks of incorrect OR.

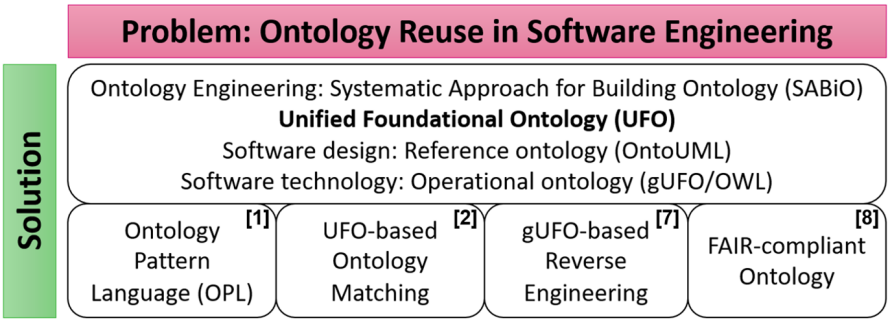

Figure 2: Four pillars to improve ontology reuse in software engineering

4.2. Theoretical Foundations

From the O literature and our prior work on the four pillars based on foundational ontologies - ontology patterns [1], ontology matching [2], reverse engineering [7] and FAIR for ontologies [8, 9] - we propose an O theory model enhanced by the ontology-driven conceptual modeling practice [15] embedded in ontology-driven SE. The model will be concertized through a set of O guidelines, which addresses R1 requirement, with detailed instructions on the characteristics needed by ontologies designed for reuse and gives precise instructions for implementing the framework services.

In this theory, we advocate that the O problem is directly associated with the semantic interoperability problem, and can be alleviated by systematically applying these four pillars while considering methodological aspects of software and ontology engineering [3]. In addition, this theory will include a specific terminology definition to make the semantics behind concepts like using, applying, implementing and reusing ontologies clear. This precise definition will provide the basis for O quality metrics and how to measure reuse, addressing R2 requirement.

Furthermore, we intend to design and develop the specifications on how to apply ontology patterns at design-time by following an OPL, and on how to apply UFO to improve ontology matching at runtime. Similar to the well-known Ontology Design Pattern (ODP) approach, we propose to integrate OPL within an ontology engineering methodology (like SABiO) to improve both reference and operational ontology design. To this aim, we address three major challenges. First, the full set of patterns within OPL theory [1] requires further experimentation and prototyping to understand their usefulness. Second, since we are proposing the use of UFO-based ontology matching, it is required to investigate how these techniques should be integrated in a systematic way. Third, there is a need to integrate ontology matching with OPL while offering ontology pattern management. To address these challenges, we intend to leverage on our approach that infers ontological categories of OWL-based ontologies through foundational rules [7] and its impact on OPL management.

4.3. Framework Services

Table 1 presents the list of the services described in this section that aim at addressing R3 requirement of the framework. We intend to advance the tooling support for the following

Table 1
Onto4Reuse services according to Software Engineering life cycle

Software Engineering phase	Service
Requirements (specification)	Ontology Modularization
Requirements (specification)	Dictionary and Competency Question Design
Design (specification)	Reference Ontology Design Patterns
Design (specification)	Reference Ontology Matching
Coding (implementation)	Operational Ontology Design Patterns
Coding (implementation)	Operational Ontology Matching
Testing and documentation	Reusability Measuring
Testing and documentation	Ontology Verification, Validation and Documentation

ontology engineering phases, according to SABiO. We propose mapping SE phases to the Ontology Engineering phases as follows. Requirement analysis is mapped to ontology purpose and requirements elicitation. Software design is mapped to ontology capture, formalization and design, which implies the adoption of reference ontologies for conceptual modelling. Then, software coding (implementation) is mapped to ontology implementation, and finally software testing is mapped to operational ontology testing.

First, on ontology purpose and requirements elicitation, we intend to improve sub-ontology identification (ontology modularization) by leveraging on Domain-Driven Design (DDD) practice, a popular approach often used for microservices architecture, and data mesh topology modeling to decompose domains according to bounded contexts and aggregates. Second, on ontology capture and formalization, we intend to improve conceptual modeling through formalizing competency questions and a dictionary with searching functions for finding and accessing terms of existing ontology indexes (e.g., Linked Open Vocabulary) leveraged by Large Language Models. In addition, we intend to operationalize the management of patterns with OPL and a pattern recommendation service according to competency questions for reference ontologies. Third, in architecture design, we intend to improve the analysis of non-functional requirements for choosing the appropriate technology for the operational ontology, considering the impact of O in performance, scalability, reliability and security. Fourth, on operational ontology implementation, we intend to cover the ontology patterns along with an ontology matching approach to generate the operational ontology. The Scior reverse engineering approach will play a relevant role in grounding the pre-selected ontologies (from the dictionary) in gUFO, allowing the further execution of the operational ontology matching approach based on UFO, generating the required alignments from the ontology under construction to the reused ones. Finally, on ontology testing and documentation, we intend to develop a service to measure reusability, and the adoption of popular approaches for verification and validation (e.g., Alloy Analyzer[10] and OOPS![11]), and documentation (e.g., WIDOCO[12]) services.

[10]https://alloytools.org/
[11]https://oops.linkeddata.es/
[12]https://github.com/dgarijo/Widoco

5. Onto4Reuse Framework Application Cases

The Onto4Reuse framework will be tested in two application domains, as follows.

5.1. Infrastructure Condition Assessment

The first application domain is infrastructure condition assessment, a specific engineering-heavy task in the domain of civil engineering and public asset management (roads, bridges, tunnels). This use case focuses on supporting the prioritization process of the civil structure portfolio of the Dutch Road Authority (Rijkswaterstaat) prior to their replacement and renovation decision-making. Rijkswaterstaat manages approximately 6600 assets, 1700 of them built before 1960. A significant proportion of their portfolio needs to respond to future demands and Rijkswaterstaat needs to understand which objects need urgent maintenance and for what underlying reasons. They struggle to gain an overview of the condition of assets in their civil structures.

The data indicating potential failures are available, and collected periodically by various departments. The data are registered through technical observations and include degradation processes, damage, failure events and planned measures connected to the physical decomposition of the structures. Yet they are locked in various software and departments, following different ontologies and no substantial indication of reuse. Nevertheless, various domain ontologies and rule sets are available for civil infrastructures. Besides the IFC ontology, which recently also covers infrastructure assets, the Damage Topology Ontology (DOT) is available, which aims to classify damages on infrastructure. Further, the Bridge Topology Ontology (BROT) provides definitions of bridge constructions including aggregated zones and components as well as their topological relations. Even though this ontology supports a generic modeling approach for any bridge type, it does not correspond to the NEN Standard 2667-4 that both public and commercial companies in the Netherlands need to comply with.

Rijkswaterstaat also has its own ontology (RWS-OTL) and a SHACL rule set per organizational process. It is not clear how much of these ontologies include O, or how reusable this RWS-OTL is. In the above context, and with the developed software and guidelines in the Onto4Reuse Framework, this case study will focus on 1) evaluating the available ontologies (RWS-OTL and open ontologies) and improving them in terms of reuse (two directions), 2) evaluating and re-creating object repositories, as knowledge graphs, with infrastructure data using the improved ontologies and 3) providing an explainable, traceable, rapid reasoning service to identify major intervention needs based on available and unified data.

5.2. Modular Construction Engineering

This second use case is in the domain of modular construction of housing. In the Netherlands we should build an additional 981.000 houses by 2030[13] and retrofit around 50.000 houses yearly in 2023 reaching up to 150.000 houses yearly in 2030[14], making them energy neutral. Current productivity and production rates are lacking due to the fact that construction is a fragmented and project-based industry [19, 20], meaning that the wheel is invented over again for each

[13] https://www.rijksoverheid.nl/actueel/nieuws/2023/07/12/woningbouwopgave-stijgt-naar-981.000-tot-en-met-2030
[14] https://www.eib.nl/wp-content/uploads/2024/05/Klimaatambities-in-de-gebouwde-omgeving.pdf

project. Modular construction is seen as a promising approach for increasing productivity in the industry by supporting a process-based approach emphasizing the reuse of information and designs [21]. The industry knows various existing ontologies and schemas supporting information sharing and interoperability including the IFC schema, the BOT ontology, and several of the available product classification ontologies (BEO, MEP, BPO, etc.). Yet, these existing ontologies and schemas share similar concepts and relations, making it challenging to reuse or map them [10]. Therefore, this use case aims to develop a modular construction engineering ontology, by reusing the already existing ontologies and shema's in the industry. This use case thus is expected to be a test case for the plethora of ontologies that is available though the Linked Building Data (LBD) ontology network approach. In a second step, a practical case study will evaluate the ontology, using the ontology verification support service.

This use case follows these steps: (1) Formalizing multi-level and multi-criteria design rules to design modular construction elements and use them in broader construction engineering practices; (2) Digitization of expert knowledge through a) (re)use of ontologies that formalize the information requirements of standardized construction elements and their relation to other construction engineering practices, and b) aligning with existing best practices in the modular construction engineering domain outside the scope of ontology engineering. (3) Creating a knowledge graph for a modular construction engineering project by instantiating the modular construction engineering ontology and exploiting available data from different heterogeneous sources, including the sources mentioned earlier in this section. (4) Developing a design and decision support system based on a Digital Twin for modular construction engineering built on top of the knowledge graph.

6. Conclusion

Ontologies are essential in knowledge-intensive SE, but their reuse is challenging, leading to redundancy and wasted resources, especially in the system engineering field. In domains like construction and infrastructure, structured data and ontologies are crucial for annotating and classifying assets. This paper proposes a research agenda to tackle O issues, focusing on two use cases in system engineering: infrastructure asset management and modular building construction. Here we introduce the Onto4Reuse framework, aiming to assist software engineers in designing and incorporating ontologies by leveraging existing ones efficiently. The framework is built on four pillars: OPL for design patterns, improved ontology matching with UFO, semi-automatic reverse engineering to ground existing operational ontologies in OWL, and adherence to the FAIR data principles. These techniques support the development of ontology-driven systems like knowledge graphs and digital twins, offering services for specification, implementation, and testing/documentation.

The core contribution of this paper is the Onto4Reuse framework that is built on top of the aforementioned four pillars and implemented through a set of services that aim at supporting software engineers. Among the main lessons learned from this preliminary work is the identification of open issues to be addressed by the research agenda proposed here. Among the main limitations of this research agenda is that it requires further in-depth analysis of the O problem, which requires a systematic literature review in the topic and a field research. We aim

at addressing this issue through the first element of Onto4Reuse framework, in the O Theory and Framework Requirements.

Another open issue is the lack of a strict and transparent definition (or a model) of what is O, considering terms that are often used in ontology-driven SE, including (re)using, applying and implementing ontologies. Therefore, the main future work is to use such a model to ground the O theory towards O guidelines. Following the Onto4Reuse reference architecture, the next steps are: ensuring that requirements for the framework from the use cases are well elicited, specifying how OPL can be integrated with ontology matching, developing the services, applying them in the real-life engineering cases and assessing how efficient and effective the Onto4Reuse Framework is.

References

[1] R. de Almeida Falbo, M. P. Barcellos, F. B. Ruy, G. Guizzardi, R. S. S. Guizzardi, Ontology pattern languages, in: P. Hitzler, A. Gangemi, K. Janowicz, A. Krisnadhi, V. Presutti (Eds.), Ontology Engineering with Ontology Design Patterns - Foundations and Applications, volume 25 of *Studies on the Semantic Web*, IOS Press, 2016, pp. 133–159. URL: https://doi.org/10.3233/978-1-61499-676-7-133. doi:10.3233/978-1-61499-676-7-133.

[2] C. Trojahn, R. Vieira, D. Schmidt, A. Pease, G. Guizzardi, Foundational ontologies meet ontology matching: A survey, Semantic Web 13 (2022) 685–704. URL: https://doi.org/10.3233/SW-210447. doi:10.3233/SW-210447.

[3] J. L. Rebelo Moreira, The role of interoperability for digital twins, in: T. P. Sales, S. de Kinderen, H. A. Proper, L. Pufahl, D. Karastoyanova, M. van Sinderen (Eds.), Enterprise Design, Operations, and Computing. EDOC 2023 Workshops, Springer Nature Switzerland, Cham, 2024, pp. 139–157.

[4] V. A. Carriero, M. Daquino, A. Gangemi, A. G. Nuzzolese, S. Peroni, V. Presutti, F. Tomasi, The landscape of ontology reuse approaches, ArXiv abs/2011.12599 (2020). URL: https://api.semanticscholar.org/CorpusID:227162282.

[5] M. Halper, L. N. Soldatova, M. Brochhausen, F. S. Maikore, C. Ochs, Y. Perl, Guidelines for the reuse of ontology content, Appl. Ontology 18 (2023) 5–29. URL: https://api.semanticscholar.org/CorpusID:256167882.

[6] M. Katsumi, M. Grüninger, Choosing ontologies for reuse, Appl. Ontology 12 (2017) 195–221. URL: https://api.semanticscholar.org/CorpusID:42383519.

[7] P. P. F. Barcelos, T. P. Sales, E. Romanenko, J. P. A. Almeida, G. Engelberg, D. Klein, G. Guizzardi, Inferring ontological categories of owl classes using foundational rules, in: Formal Ontology in Information Systems, 2023. URL: https://api.semanticscholar.org/CorpusID:266848416.

[8] R. Çelebi, J. L. R. Moreira, A. A. Hassan, S. Ayyar, L. Ridder, T. Kuhn, M. Dumontier, Towards fair protocols and workflows: the openpredict use case, PeerJ Computer Science 6 (2020). URL: https://api.semanticscholar.org/CorpusID:224772139.

[9] T. P. Sales, P. P. F. Barcelos, C. M. Fonseca, I. V. Souza, E. Romanenko, C. H. Bernabé, L. O. B. da Silva Santos, M. Fumagalli, J. Kritz, J. P. A. Almeida, G. Guizzardi, A fair

catalog of ontology-driven conceptual models, Data Knowl. Eng. 147 (2023) 102210. URL: https://api.semanticscholar.org/CorpusID:260784004.
[10] P. Pauwels, K. McGlinn, Buildings and Semantics: Data Models and Web Technologies for the Built Environment (1st ed.)., CRC Press, https://doi.org/10.1201/9781003204381, 2022.
[11] D. R. Shelden, P. Pauwels, P. Pishdad-Bozorgi, S. Tang, Data standards and data exchange for construction 4.0, 2020. URL: https://api.semanticscholar.org/CorpusID:213889610.
[12] P. Sowiński, K. Wasielewska-Michniewska, M. Ganzha, M. Paprzycki, C. Bădică, Ontology reuse: the real test of ontological design, in: New Trends in Software Methodologies, Tools and Techniques, 2022. URL: https://api.semanticscholar.org/CorpusID:248562888.
[13] G. F. Schneider, Automated ontology matching in the architecture, engineering and construction domain - a case study, in: LDAC, 2019. URL: https://api.semanticscholar.org/CorpusID:197642195.
[14] S. Stolk, W. D. Lubbers, F. Braakman, S. Weitkamp, Ontologies and json-ld at tennet: The use of linked data on eu-303 projects, in: LDAC@ESWC, 2022. URL: https://api.semanticscholar.org/CorpusID:253000607.
[15] G. Guizzardi, A. B. Benevides, C. M. Fonseca, D. Porello, J. P. A. Almeida, T. P. Sales, Ufo: Unified foundational ontology, Appl. Ontology 17 (2021) 167–210. URL: https://api.semanticscholar.org/CorpusID:244875087.
[16] P. I. Nakagawa, L. Ferreira Pires, J. L. Rebelo Moreira, L. Olavo Bonino, Towards semantic description of explainable machine learning workflows, in: 2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW), 2021, pp. 236–244. doi:10.1109/EDOCW52865.2021.00054.
[17] K. I. Kotis, G. A. Vouros, D. Spiliotopoulos, Ontology engineering methodologies for the evolution of living and reused ontologies: status, trends, findings and recommendations, The Knowledge Engineering Review 35 (2020) e4. doi:10.1017/S0269888920000065.
[18] J. L. R. Moreira, L. F. Pires, M. van Sinderen, L. Daniele, M. Girod-Genet, Saref4health: Towards iot standard-based ontology-driven cardiac e-health systems, Appl. Ontology 15 (2020) 385–410. URL: https://api.semanticscholar.org/CorpusID:221356028.
[19] K. Fergusson, S. U. C. for Integrated Facility Engineering, S. U. D. of Civil Engineering, P. Teicholz, Impact of Integration on Industrial Facility Quality, CIFE technical report, Stanford University, 1993. URL: https://books.google.nl/books?id=JDMEAAAAIAAJ.
[20] A. Adriaanse, Bruggen bouwen met ICT, University of Twente, Netherlands, 2014. Rede uitgesproken bij de aanvaarding van het ambt van hoogleraar Bouprocesintegratie ICT aan de fac. Construerende Technische Wetenschappen van de Universiteit Twente op donderdag 9 oktober 2014.
[21] C. P. Nick Blismas, A. Gibb, Benefit evaluation for off-site production in construction, Construction Management and Economics 24 (2006) 121–130. doi:10.1080/01446190500184444.