EG-ICE 2025
GLASGOW

Interpreting SysML Diagrams in conjunction with OWL Ontologies for Semantic Reasoning and Inference

BENEDICT HARDER[1], SEBASTIAN ESSER[1], ANDRÉ BORRMANN[1]

Technische Universität München, Georg Nemetschek Institute

ABSTRACT:
This paper investigates how to enrich SysML (Systems Modeling Language) models by integrating them with the semantic reasoning capabilities of OWL to find out whether model analysis can be improved by inferring new relationships and detecting inconsistencies. To answer this, we propose a methodology that systematically translates SysML block definition and requirements diagrams into OWL and then represents the resulting data in an RDF graph database. The method infers new topological and requirement-based insights, demonstrating improved model accuracy and deeper information extraction. This result is significant for engineers and systems designers, enhancing decision-making and error detection, especially within the context of modular design and cyber-physical systems. Moreover, the findings pave the way for further research into hybrid modeling approaches, suggesting promising avenues for integrating formal semantics into model-based systems engineering.

KEYWORDS:
MBSE, SysML, OWL, Ontologies, Reasoning

1. INTRODUCTION

As engineering knowledge has significantly evolved over the past decades and centuries, various methods for capturing, structuring, and utilizing this knowledge have emerged. These methods have been shaped by the increasing complexity of engineering domains and the growing need for precise and standardized ways of representing information.

On one hand, formal ontologies, such as those built using the Web Ontology Language (OWL), provide a structured framework for encapsulating domain-specific knowledge. OWL achieves this by defining a formal schema that dictates how data should be structured and how entities within a domain relate to one another (Gruber, 1993). Through logical reasoning mechanisms, OWL enables the inference of new knowledge based on existing relationships and constraints, making it a powerful tool for knowledge representation and automated decision-making.

On the other hand, descriptive modeling languages, such as the Systems Modeling Language (SysML), offer a different yet complementary approach. SysML, which is a specialized dialect of the Unified Modeling Language (UML), is widely used in systems engineering to model complex systems, their interdependencies, and their requirements (Fowler, 2003). By providing a rich set of diagrammatic representations, SysML allows engineers to describe system architectures, requirements, behaviors, and interactions in a structured and comprehensible manner, facilitating communication among stakeholders and ensuring consistency and traceability in system design.

Harder, B., Esser, S., & Borrmann, A. Interpreting SysML diagrams in conjunction with OWL ontologies for semantic reasoning and inference. In A. Moreno-Rangel, & B. Kumar (Eds.), EG-ICE 2025: AI-Driven Collaboration for Sustainable and Resilient Built Environments Conference Proceedings University of Strathclyde Publishing. https://doi.org/10.17868/strath.00093300

Historically, OWL and SysML have been developed and applied in largely separate contexts, with limited collaborative usage. This is somewhat surprising given that both languages share common goals: they seek to define relationships between data objects, enforce structural consistency, and support the analysis of complex systems. However, their primary differences lie in how they achieve these objectives. OWL, rooted in semantic web technologies, excels in defining and enforcing logical constraints, enabling automated reasoning over large datasets. In contrast, SysML prioritizes explicit system representation, making it a preferred choice for human-centric system modeling and engineering design (Delligatti, 2013).

Despite these distinctions, there is significant untapped potential in combining these two methodologies. This paper proposes that the reasoning capabilities of OWL can enhance the analytical rigor of SysML-based system models, while SysML's structured, domain-specific representations can provide a more tangible and visual means of leveraging OWL's inferencing capabilities. To date, however, little research has explored the integration of these two approaches.

This paper aims to bridge this gap by investigating the potential synergy between the Web Ontology Language and the Systems Modeling Language. Specifically, it explores how OWL's reasoning capabilities can be combined with SysML's descriptive and visual modeling strengths to enhance interdisciplinary knowledge transfer and improve engineering workflows. To demonstrate the practical implications of this integration, this study applies these concepts in case studies within the domain of modular bridge construction, showcasing how a hybrid approach can facilitate better system analysis, design verification, and automated consistency checks. This is especially relevant in the context of automated manufacturing using Cyber-Physical Systems (CPS), which have become more and more prevalent in the modular construction industry (Anumba *et al.*, 2021).

2. BACKGROUND

2.1 MODEL-BASED-SYSTEMS-ENGINEERING AND THE SYSTEMS MODELING LANGUAGE

Systems Engineering is defined by (Sillitto *et al.*, 2019) as follows: "A transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using principles and concepts, and scientific, technological, and management methods." The core of this approach is engaging across multiple engineering disciplines and conglomerating diverse fields of knowledge. Model-Based Systems Engineering (MBSE), on the other hand, captures this same way of thinking and adapts it into a model-based methodology, ensuring consistency across conceptualization, development, and operation. By using a fully integrated model as the underlying basis for designing and developing the system, organizations can improve traceability, enhance collaboration across disciplines, and streamline verification and validation processes throughout the entire lifecycle (Friedenthal, Moore and Steiner, 2014). This, in general, entails moving away from document-based information, a similar concept to what Building Information Modeling (BIM) is to the Architecture, Engineering, and Construction (AEC) industry (Borrmann *et al.*, 2021).

MBSE can be divided into three pillars: (1) The method of system development, (2) the language used for system development, and (3) the tool used for the former two (Delligatti, 2013). While each pillar is independent from the other two, common standards have been established in the industry. Methods such as the Object-Oriented Systems Engineering Method (OOSEM) and MagicGrid have established themselves (Estefan, 2008), while the Systems Modeling Language (SysML) has been the de facto standard since its inception in 2003 (Friedenthal, Moore and Steiner, 2014). There exists a plethora of tooling with which both method and language can be used, with the software products Cameo Systems Modeler [1] and Enterprise Architect [2] being widely used in the industry.

With its model-based approach, consistency, and traceability, MBSE has become a powerful tool to model real-world systems, with it largely being used in the automotive and aerospace industries (Akundi *et al.*, 2022). While there have been attempts at adapting MBSE for the AEC domain, this field has so far been largely untouched, with few exceptions. As previously mentioned, the Systems Modeling Language has established itself as the de facto standard modeling language for MBSE. Being a dialect of the Unified Modeling Language (UML), it makes use of common object-oriented concepts, albeit with slightly different vocabulary (Delligatti, 2013). It encompasses a wide range of diagram types, all serving the purpose of declaratively describing the requirements, structure, behavior, and parametrics of the system model, also known as the four pillars of SysML (Hause, 2011). Many of the diagram types are directly taken from the UML 2.0 specification, with some additional diagram types being added specifically for SysML,

[1] https://www.3ds.com/products/catia/no-magic/cameo-systems-modeler

[2] https://sparxsystems.com/

The purpose of the diagrams is to not only provide a visual representation of (a part of) the model but also serve as a window for the system developer to view and make changes to the underlying model. Various software tools such as the ones previously mentioned assist developers in maintaining this consistency by constructing a specification-compliant model based on the drawn diagrams.

MBSE, in conjunction with SysML, has proven to be a reliant way of modeling systems accurately and consistently. Adoption into the AEC domain, though, has remained limited even though this method has been established in the automotive and aerospace industries. This for sure has a multitude of reasons, for example the more geometry-driven way of working in construction and building engineering (Eastman, 2011). The integration of stakeholder requirements and a higher-level model structure directly into the system model itself, though, can prove to be something the AEC domain can profit from.

2.2 SEMANTIC WEB AND ONTOLOGIES

The concept behind the semantic web has been developed in large parts by Tim Berners-Lee, where information interchange across the web should happen in a standardized yet flexible way. Core parts of realizing this concept are the Resource Description Framework (RDF) and associated ontologies. RDF serves as a way of exchanging interconnected information in the form of a graph, which makes associating and linking information simple by transferring data packages in the form of these RDF graphs (Lassila, Hendler and Berners-Lee, 2001).

Ontologies, on the other hand, serve on a meta-level to RDF graphs by providing the option to define classes, cardinalities, and types of relationships between said classes (Allemang and Hendler, 2011). These capabilities mirror many concepts from the object-oriented paradigm, though it does need to be stated that as opposed to languages like UML (which are also object-oriented), ontologies do not subscribe to the closed-world assumption, which assumes that things are not inherently untrue because they have not been explicitly defined. For this reason, ontologies and their implementations, such as the Web Ontology Language (OWL), are often employed in contexts where inference is desired rather than strict adherence to a fixed schema (OWL, 2024).

SysML, being a dialect of UML, generally adheres to the closed-world assumption (Friedenthal, Moore and Steiner, 2014), meaning that everything that is assumed to be true must also be explicitly stated. This, of course, lies in contrast with what was previously stated about OWL. Yet, this does not mean that these two languages are incompatible with each other. Rather, they may complement each other, with SysML being used for formal product/system development and OWL reasoning and inferencing capabilities being used for semantic rigorousness and analysis. OWL has been extended to further its reasoning and inferencing capabilities through, for example, the likes of the Semantic Web Rule Language (SWRL), which leverages patterns to create new relationships in a Semantic Web Representation (Horrocks *et al.*, 2004).

2.3 SPECIFIC RELATED RESEARCH CONTRIBUTIONS

There have been a number of research contributions over the past years pertaining to this topic. The parallels between SysML and OWL are reason enough to examine them closer and find out where they can extend and complement each other. Graves, H., 2009, for example, proposes translating a restricted subset of SysML Block Diagrams into OWL2 Knowledge Bases (KBs) to enable formal reasoning about system designs. This is done by defining a model-theoretic semantics for Block Diagrams and mapping elements like blocks, parts, and cardinality constraints into OWL2 classes, properties, and axioms. The translation preserves the intended semantics of the SysML diagrams, allowing the use of OWL2 tools for design validation and consistency checking. The result is a partial integration of SysML's graphical modeling with OWL2's formal logic, enabling structural system representations to be used for automated reasoning.

Jenkins and Rouquette, 2012 describe a method to integrate SysML and OWL by transforming SysML models into formally defined OWL ontologies. It uses custom OWL ontologies to define systems engineering concepts and embeds them into SysML profiles using QVTo transformations. These profiles guide the construction of semantically rigorous SysML models, which can then be transformed back into OWL for formal reasoning, validation, and query answering. The approach enables bidirectional translation and semantic analysis across graphical (SysML) and logical (OWL) modeling environments.

Wardhana, Ashari and Sari, 2020 on the other hand, presents a model that automatically transforms SysML Requirement Diagrams into OWL ontologies. It uses XMI serialization as input and applies transformation rules and an algorithm to generate OWL files in RDF/XML format. The approach maps SysML elements (e.g., requirements, dependencies, containments) to OWL components like classes, properties, and subclass relationships. The result is a machine-readable ontology that can be visualized and validated in software tools like

Protégé[3], reducing manual effort and improving consistency.

Additionally, graph data structures to represent AEC models have been heavily researched in recent years and promise to be a suitable method for representing buildings digitally, enabling version control (Esser, Vilgertshofer and Borrmann, 2023) and consistency maintenance (Wang, Ouyang and Sacks, 2023), making graph-based solutions for engineering consistency relationship inferencing even more prevalent.

2.4 RESEARCH GAP

Most, if not all, contributions make use of a very direct SysML to OWL mapping specifically to exploit either the reasoning mechanisms to infer new knowledge or trigger validation engines to ensure data model consistency. One area, though, in which little research has so far been conducted is the use of ontologies to not only validate the system model on the datamodel level, but also on the instance level. In this case, the system model (SysML diagrams) would serve as a way to declaratively describe model requirements and structure. In contrast, a tangible BIM or Computer Aided Design (CAD) model is typically developed in parallel, describing the specific instances of components forming the built asset in question. Ontologies can then be derived from the SysML model as a means to enable palpable interaction with the BIM model through reasoning, validation, and relationship inference.

With this in mind, the following research objectives have been outlined:

- Develop a method of deriving OWL axioms and constraints from SysML requirement and block definition diagrams by means of a comprehensive mapping.
- The usage of said ontologies to validate BIM models in regards to structure and requirement satisfaction.
- Inference of new relationships within the BIM model for structural analysis.

3. METHODOLOGY

The method to achieve the above-mentioned objectives can be split into the following parts: (1) Converting a BIM model into an RDF representation, (2) Developing a modeling technique for semantically rich and consistent SysML model diagrams, and (3) serializing these diagrams into OWL ontologies in the form of turtle (.ttl) files, and (4) applying said ontologies to the RDF representation, leading to structural topology inference and requirement satisfaction. The following explanations will frequently be illustrated with examples from modular bridge construction, such as the one shown in Fig. 1:

Figure 1: Sample module arch bridge, containing a road surface and girders (superstructure), as well as foundations, abutments, and arches (substructure).

The process, as detailed in Fig. 2, begins by converting an existing model of the system into an RDF graph representation. This transformation is performed using widely recognized ontologies, such as the Building Ontology Topology (BOT), which provides standardized terms and relationships for representing building-related data.

Simultaneously, a set of SysML (Systems Modeling Language) diagrams is created. These diagrams serve to formally define the structure of the building by illustrating its hierarchical decomposition into components. They also explicitly model key relationships, such as the "supports" relationship between load-bearing elements, and specify the relevant system requirements that the building must fulfill.

From the information encoded in these SysML diagrams, corresponding ontological representations are derived. These include OWL axioms, logical constraints, and SWRL rules. These ontological elements serve two main purposes: first, they enable the automated inference of structural topology information, such as which components support others, and second, they allow for the inference of "satisfy" relationships, which link specific components or subsystems to the system requirements they fulfill. The RDF representation, including the self-defined axioms and SWRL rules, are applied together in a graph database with reasoning capabilities.

3.1 RDF CONVERSION

For converting from a common building data format, such as the Industry Foundation Classes (IFC), to an RDF representation, existing methods can be applied out of the box, such as the IFC to Linked Building Data (LBD) converter proposed by (Bonduel *et al.*, 2018). This conversion tool adheres to the LBD ontology created by the Linked Building Data Community Group (W3C-LBD-CG, 2024). This serves as a foundation for moving the BIM model to an RDF representation for further analysis and processing. The existence of a clearly defined and

[3] https://protege.stanford.edu/

widely accepted building ontology is not only consistent with common engineering knowledge but also makes interfacing with custom ontologies easy by defining which OWL classes are equivalent later on. Of course, other appropriate ontologies could also be used here.

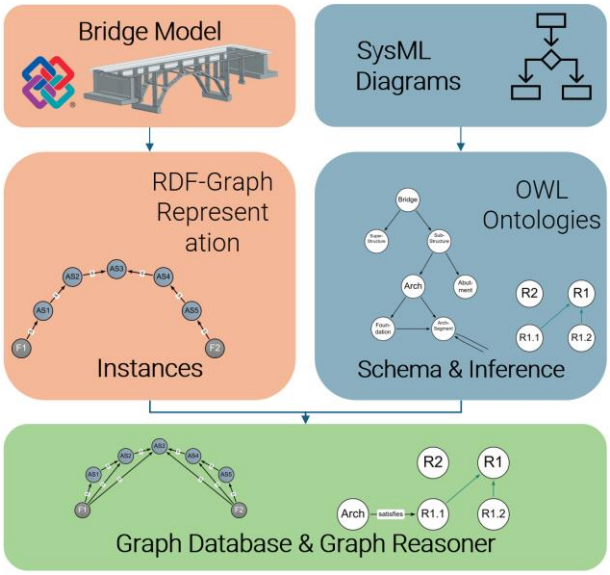

Figure 2: Overall workflow and application of the method.

3.2 SYSML DIAGRAMS & OWL CONVERSION

SysML diagrams allow for a wide range of expressiveness through various diagram types. For now, this paper focuses on block definition diagrams (bdd) (for an example, see Figure 3) and requirement diagrams (req) (Figure 4), as these two types form the core of requirements and structure definition, and allow for the semantic articulation necessary for the conversion into ontologies by providing methods of modeling classes, requirements, and their relationships. A bdd mainly consists of blocks and different types of associations between these blocks, such as composition and generalization. With this, it is the equivalent of a UML class diagram with a different nomenclature. While blocks allow for the definition of attributes and methods (called values and operations), the main features that will be made use of are the “pure” block definitions as well as the associations between them.

3.2.1 Block Definition Diagrams

At first, the structural composition must be defined using block diagrams. Structural composition refers to a top-down definition of a decomposition from the building as a whole down to single building components. What this achieves is a rigorous formalization of how different building components relate to each other in terms of composition. Block definition diagrams can model composition and generalizations, which in turn means that subclasses and inheritance are also possible. This is especially convenient for building components that share semantic meaning, such as modular components (i.e., from concrete), which can all specialize from a general “Module” block or something similar. Furthermore, general associations (i.e. ones with no compositional meaning) are used to model structural topology. For example, through a general association with the label “supports” or “is supported by”, the load-bearing relationship between two building components, such as two modules, can be articulated. A non-exhaustive example of such a block definition diagram can be seen in Fig. 3.

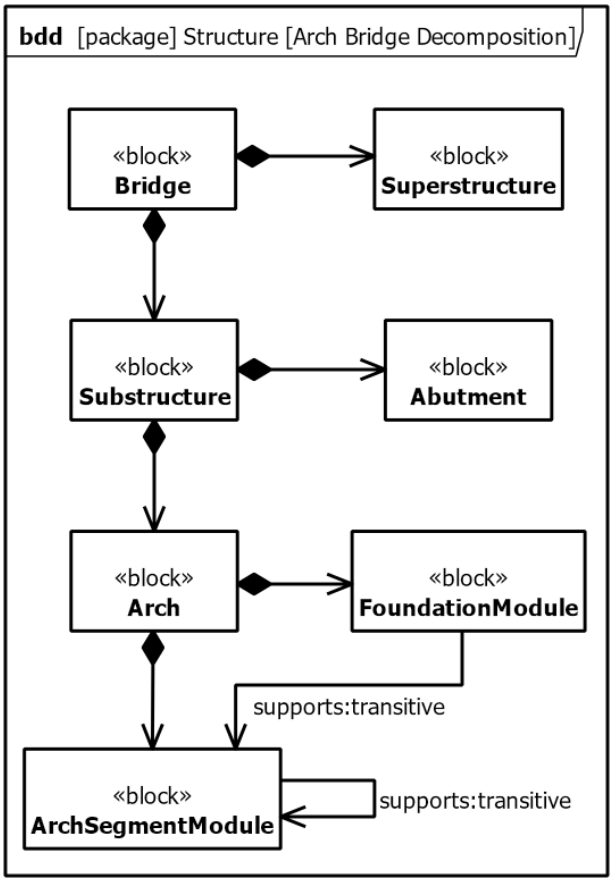

Figure 3: Sample block definition diagram (bdd) of a modular arch bridge, showcasing a non-complete decomposition down to individual modules. Between the modules, general associations indicating structural support are modeled with a “transitive” label.

3.2.2 Requirement Diagrams

The diagram definitions do not stop here, though. A central element of MBSE has been the continual process of modeling and translating stakeholder needs into system requirements. The formal definition and requirement traceability are key areas where BIM models can be approved (Casillas and Howe, 2013; Valdes *et al.*, 2016). Since this approach focuses on structural topology (and its

integrity), a requirements diagram, in our case, must articulate a structural requirement, i.e., "The bridge substructure must be structurally sound" with further requirements which refine it ("The arch segments must transmit the loads to the foundation"). An example of a requirements diagram can be seen in Fig. 4:

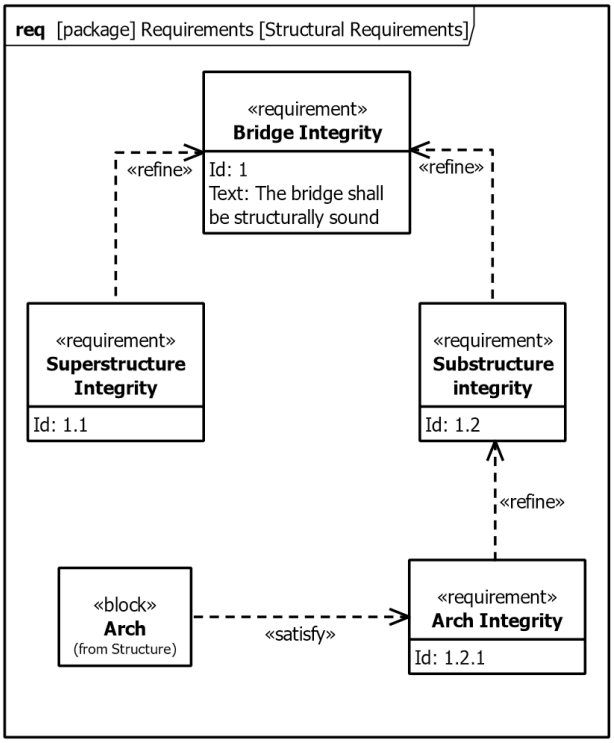

Figure 4: (Non-complete) example of a requirements diagram (req). Requirement entities are defined through a name, Id, and a textual description (sample description given in requirement No. 1). Each requirement can (or must) be satisfied by a system component, in this case, the "Arch" component taken from the previous diagram.

A key feature of requirements in MBSE and SysML is the possibility of satisfying them with system components defined in, for example, block definition diagrams. This way, requirement satisfaction remains traceable throughout the system lifecycle. This would usually be updated over the course of the model's development. This paper, however, aims to infer the "satisfaction" relationships from generated ontologies and constraints. To know which components are necessary to satisfy the requirement, though, we must formally define them inside the requirements diagram to properly convey this in the ontology later on.

3.2.3 SysML - OWL Mapping

Up to this point, an RDF representation of a bridge IFC model has been developed using the LBD ontology, accompanied by a set of block definition and requirements diagrams describing the system architecture and associated requirements. Now, a mapping between the system model and OWL ontologies is necessary. A large part of the SysML language components can be easily mapped to specific OWL features, as seen in Table 1:

Table 1: SysML to OWL mapping.

SysML	OWL
Block	Class
Generalization	SubClassOf
Composition	Object Property
Association	Object Property
Attribute	Data Property

Both composition and general association can be mapped to OWL's object property. The difference in semantic meaning of these associations can be modeled using fitting labels, such as a "*is part of*" label.

3.2.4 Structural Inference

A simple 1-to-1 mapping between these two languages may be a good start to getting a conversion going, but to achieve the main goal of this research, a deeper semantic translation must happen. OWL features axioms and constraints that enable the semantic reasoning we aim to achieve. One example is the transitivity of relationships. Transitivity states that if a relationship *r* between two pairs *(A→B)* and *(B→C)* exists, the relationship *(A→C)* also exists. Transitivity is an integral part of reasoning mechanisms that infer new relationships. This can be made use of in a structural example: Picture an arch bridge whose arch consists of multiple arch segments, two of which are supported by a foundation, as seen in Fig. 5:

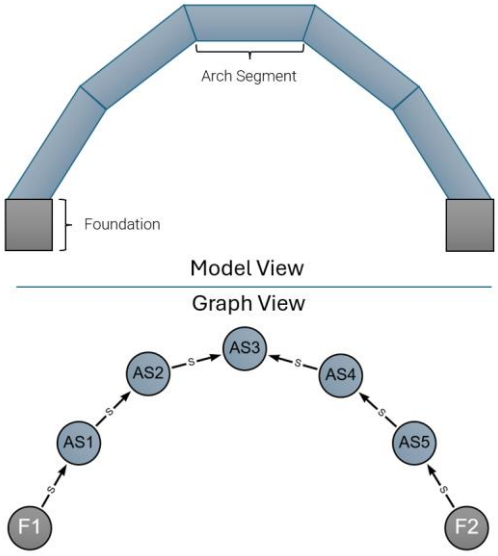

Figure 5: 2D-View of a bridge-models arch and the corresponding (sample) graph representation, including the "supports" relationships (s). (AS) represents arch segment nodes, (F) represents foundation nodes.

Should the "*supports"* relationship be declared transitive, new relationships could automatically be inferred between the arch segments so far not semantically connected to a foundation. The resulting graph structure after applying the reasoner, including the OWL constraints, can be seen in Fig. 6:

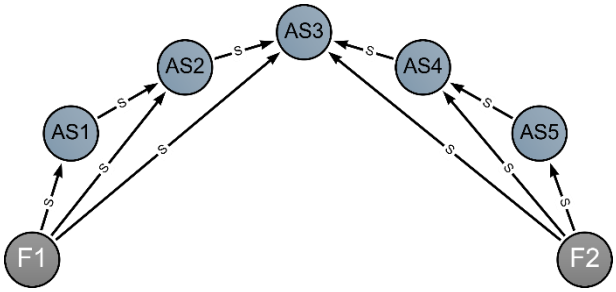

Figure 6: The same graph representation after invoking the reasoning mechanisms based on the derived axioms. Additional relationships between the arch segment nodes and foundation nodes have been created, indicating that loads from an arch segment can be redirected to the foundations, even if they are not directly connected to each other.

If one of the arch segments is not correctly connected to its adjacent modules, it means that its loads can never be properly distributed to the foundations. In order to convey this transitivity of the relationships inside the SysML diagrams, we can label the associations accordingly with "*supports:transitive"*, as can be seen in Fig. 1.

With this approach, basic structural analysis from can be conducted without the need for cumbersome finite element methods or manual checking. Additionally, having a direct relationship from a modular or segment to its final load-bearing element (in this case, a foundation) makes such an analysis easier and more intuitive.

3.2.5 Requirement Satisfaction Inference

We can apply a similar concept to the previously created requirements diagrams to extend this idea. As mentioned earlier, formal requirements in SysML can be linked to system components (i.e., Blocks) through a special "*satisfies*" relationship. For example, a structural integrity requirement for a bridge substructure may be satisfied by a "Substructure" block (see Fig. 1). While drawing this relationship is straightforward, the engineering behind actually satisfying the requirement must be considered. Similar to how reasoning mechanisms were used to infer structural topology, a similar process can be applied to requirements.

Assuming (1) the integrity requirement is satisfied by a single block representing the (sub)structure, and (2) this block is decomposed into all structurally relevant modules, we can infer satisfaction by verifying that all module instances are supported, e.g., by a foundation. More precisely, if the structural topology is valid, as discussed earlier, we can assume that the integrity requirement shown in Fig. 7 is fulfilled.

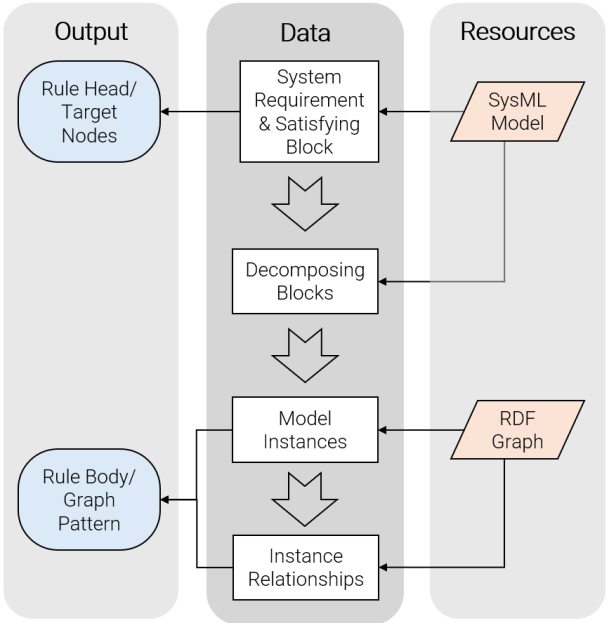

Figure 7: Workflow of deriving SWRL rules from the SysML diagrams. By retrieving the requirement component, as well as the relating system block (and its composing elements), the rule head of the SWRL rule can be derived (the target relationship that is to be created). From the composing elements, the pattern to be matched (the rule body) is determined.

Beginning from the system model (and the SysML diagrams), a requirement is chosen, together with the block that satisfies or is supposed to satisfy it. Then, the blocks that decompose the structure are fetched and checked for any supporting and transitive relationship between them, i.e., between an arch segment and a foundation. With these blocks retrieved, one can query the RDF graph for all relevant component instances of this type in the model and check for the appropriate connections. To stay in the example of arch integrity, we can check for every arch segment instance to be directly supported by at least one foundation, which we have inferred in the previous step. Now, all necessary components to construct an SWRL rule are complete.

4. IMPLEMENTATION

To implement said approach, the sample bridge shown in Fig. 1 was converted from IFC format to RDF format using the converter tool mentioned in section 3.1. The output file is in turtle format (.ttl) and follows the Building Ontology Topology (BOT). In parallel, SysML diagrams were created, which are shown in Fig. 2 and 3, respectively. Before enacting the SysML – OWL conversion, we must consider a few key points:

- While blocks may be easy to map to OWL classes, requirements are a different case and are more easily categorized into instance-level nodes.
- Concrete data instances belonging to the bridge model are not yet present.

With this in mind, we can begin creating an ontology from the SysML diagrams. This results in OWL class definitions, which decompose a modular arch bridge structure and define the transitive "support" relationships between concrete modules. Additionally, a "Requirement" class can be defined in order for requirement instances to be created from them. Subclass relationships must be defined towards the BOT, in order to ensure the general model structure:

```
:Bridge a owl:Class ;
    rdfs:subClassOf bot:Building ;
        [ a owl:Restriction ;
          owl:onProperty :decomposedBy ;
          owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
          owl:onClass :Substructure ] ,
        [ a owl:Restriction ;
          owl:onProperty :decomposedBy ;
          owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
          owl:onClass :Superstructure ] .
```

Listing 1: Sample class definition of the "Bridge" class, including subclass definitions to the bot:Building class, decomposition properties, and cardinality definitions.

We continue by retrieving the relevant element instances for our case study: ArchSegmentModules and FoundationModules. By modeling the support relationships between them, the reasoning mechanism can infer the new connections, as seen in Fig. 4 and 5.

Finally, the SWRL must be created. By using the method outlined in Fig. 6, we can derive the following SWRL rule, which checks for each ArchSegmentModule to an incoming "supports" relationship from a module. Should that be the case, the requirement instance "R1" is connected with a "satisfies" relationship:

```
[ a swrl:Imp ;
  swrl:body (
    :supports(?f1, :AS1)  ^ :FoundationModule(?f1) ;
    :supports(?f2, :AS2)  ^ :FoundationModule(?f2) ;
    :supports(?f3, :AS3)  ^ :FoundationModule(?f3) ;
    :supports(?f4, :AS4)  ^ :FoundationModule(?f4) ;
    :supports(?f5, :AS5)  ^ :FoundationModule(?f5) ;
    :Arch1(?arch) ;
    :R1(?req)
  ) ;
  swrl:head ( :satisfies(?arch, ?req) )
] .
```

Listing 2: SWRL rule to satisfy the requirement. Having retrieved the necessary model instances, it checks for each ArchSegmentModule instance to have the appropriate relationships (rule body) and then creates the new relationship for the requirement (rule head).

By enacting a SWRL-capable reasoner, such as provided by Protégé, the aforementioned new relationships are infered and/or created (as can be seen in Fig. 7), ensuring instance model consistency with the requirements defined by system model.

5. CONCLUSION & FUTURE WORK

This paper has proposed a method to enable semantic inference and reasoning in a tightly integrated manner with BIM models. By using SysML models as a central reference for semantic structure and requirement definition, engineering consistency can be validated in a clear and dependable manner with the support of knowledge representations in the form of OWL ontologies and SWRL rules.

Structural topology analysis in itself is, of course, already available in software today. Similar results to what was explained in section 3.2.4 can be achieved with existing software products such as Solibri's rule-checking capabilities. However, the ability to automatically trace and satisfy requirements inside the model graph itself by means of system engineering and requirements diagrams can prove to be paramount with ever-increasing complexity in construction projects, which makes better requirement management necessary (Karim Jallow *et al.*, 2014). Additionally, the so-far manual process of the OWL conversion can be automated to make the transition more consistent and accessible.

The presented case study can and will be extended in future research. As mentioned before, while structural topology inferencing and analysis may be useful, the automatic integration of requirements and stakeholder needs should be formalized more. The advantage of using graphs as a data representation, in this case, is that the inferencing mechanisms can extend to not only tangible digital assets but also more abstract system concepts such as stakeholder responsibilities, safety protocols, or lifecycle phases. This opens the door for holistic consistency checks that transcend the purely

physical model and instead reflect the full spectrum of system-level engineering concerns.

In conclusion, this paper has established a foundation for bridging the domains of Systems Engineering and Building Information Modeling (BIM) by integrating SysML-based systems modeling with ontology-driven approaches to structure and reasoning. By bringing requirement satisfaction, topological analysis, and consistency management into a shared formalism, we move closer to a vision of model-driven construction that is not only digital but also semantically intelligent. Future work will focus on expanding the formalization of non-structural requirements, integrating version-aware reasoning techniques, and testing the approach on larger and more heterogeneous infrastructure models. Ultimately, the goal is to make engineering intent both machine-readable and verifiable and step toward more resilient, efficient, and transparent construction processes.

ACKNOWLEDGEMENTS

We gratefully acknowledge the financial support of the Deutsche Forschungsgemeinschaft (DFG) as part of the program SPP 2187 “Adaptive modularized constructions made in flux” (Project 423969184).

REFERENCES

Akundi, A. *et al.* (2022) ‘Perceptions and the extent of Model-Based Systems Engineering (MBSE) use – An industry survey’, in *2022 IEEE International Systems Conference (SysCon)*, pp. 1–7. Available at: https://doi.org/10.1109/SysCon53536.2022.9773894.

Allemang, D. and Hendler, J. (2011) *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier.

Anumba, C.J. *et al.* (2021) ‘Cyber—physical systems development for construction applications’, *Frontiers of Engineering Management*, 8(1), pp. 72–87. Available at: https://doi.org/10.1007/s42524-020-0130-4.

Bonduel, M. *et al.* (2018) ‘The IFC to linked building data converter : current status’, in M. Poveda-Villalón, P. Pauwels, and A. Roxin (eds) *LDAC 2018 Linked Data in Architecture and Construction*. CEUR-WS.org (CEUR Workshop Proceedings), pp. 34–43.

Borrmann, A. *et al.* (2021) *Building Information Modeling - Technologische Grundlagen und industrielle Praxis*. 2nd edn. Springer Vieweg.

Casillas, R.P. and Howe, S.A. (2013) ‘Virtual Construction of Space Habitats: Connecting Building Information Models (BIM) and SysML’, in *AIAASPACE 2013 Conference and Exposition*. Available at: https://doi.org/10.2514/6.2013-5508.

Delligatti, L. (2013) *SysML distilled: A brief guide to the systems modeling language*. Addison-Wesley.

Eastman, C.M. (2011) *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons.

Esser, S., Vilgertshofer, S. and Borrmann, A. (2023) ‘Version control for asynchronous BIM collaboration: Model merging through graph analysis and transformation’, *Automation in Construction*, 155, p. 105063. Available at: https://doi.org/10.1016/J.AUTCON.2023.105063.

Estefan, J. (2008) ‘Survey of Model-Based Systems Engineering (MBSE) Methodologies’, *INCOSE MBSE Focus Group*, 25.

Fowler, M. (2003) *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional.

Friedenthal, S., Moore, A. and Steiner, R. (2014) *A Practical Guide to SysML: The Systems Modeling Language, Third Edition*, *A Practical Guide to SysML: The Systems Modeling Language, Third Edition*. Elsevier. Available at: https://doi.org/10.1016/C2013-0-14457-1.

Graves, H. (2009) ‘Integrating SysML and OWL’, in *Proceedings of the 6th International Conference on OWL: Experiences and Directions - Volume 529*. Aachen, DEU: CEUR-WS.org (OWLED’09), pp. 117–124.

Gruber, T.R. (1993) ‘A translation approach to portable ontology specifications’, *Knowledge acquisition*, 5(2), pp. 199–220.

Hause, M. (2011) ‘“Are we there yet?” Assessing Quality in Model Based Systems Engineering’, *INCOSE International Symposium*, 21, pp. 510–522. Available at: https://doi.org/10.1002/j.2334-5837.2011.tb01221.x.

Horrocks, I. *et al.* (2004) ‘SWRL: A semantic web rule language combining OWL and RuleML’.

Jenkins, S. and Rouquette, N. (2012) ‘Semantically-Rigorous Systems Engineering Using SysML and OWL’, in *International Workshop on Systems and Concurrent Engineering for Space Applications*.

Karim Jallow, A. *et al.* (2014) ‘An empirical study of the complexity of requirements management in construction projects’, *Engineering, Construction and Architectural Management*, 21(5), pp. 505–531. Available at: https://doi.org/10.1108/ECAM-09-2013-0084.

Lassila, O., Hendler, J. and Berners-Lee, T. (2001) ‘The semantic web’, *Scientific American*, 284(5), pp. 34–43.

Linked Building Data Community Group (no date). Available at: https://www.w3.org/community/lbd/ (Accessed: 1 April 2025).*OWL Web Ontology*

Language Reference (no date). Available at: https://www.w3.org/TR/owl-ref/ (Accessed: 28 March 2025).
Sillitto, H. *et al.* (2019) 'Systems engineering and system definitions', in *INCOSE*.
Valdes, F. *et al.* (2016) 'Applying Systems Modeling Approaches to Building Construction', in A. (Auburn U. Sattineni, S. (Auburn U. Azhar, and D. (Georgia T.U. Castro (eds) *Proceedings of the 33rd International Symposium on Automation and Robotics in Construction (ISARC)*. Auburn, USA: International Association for Automation and Robotics in Construction (IAARC), pp. 844–852. Available at: https://doi.org/10.22260/ISARC2016/0102.
Wang, Z., Ouyang, B. and Sacks, R. (2023) 'Graph-based inter-domain consistency maintenance for BIM models', *Automation in Construction*, 154, p. 104979. Available at: https://doi.org/10.1016/J.AUTCON.2023.104979.
Wardhana, H., Ashari, A. and Sari, A.K. (2020) 'Transformation of sysML requirement diagram into OWL ontologies', *International Journal of Advanced Computer Science and Applications*, 11(4), pp. 106–114. Available at: https://doi.org/10.14569/IJACSA.2020.0110415.