

Building Knowledge Representation for Computer Scientists: Insights from Teaching

Diellza Elshani¹ , Daniel Hernandez² , Steffen Staab^{2,3}  and Thomas Wortmann¹ 

¹Institute for Computational Design and Construction, Department for Computing in Architecture (ICD/CA), University of Stuttgart, Germany

²Department of Analytic Computing, Institute for Artificial Intelligence, University of Stuttgart, Germany

³Electronics and Computer Science, University of Southampton, United Kingdom

E-mail(s): diellza.elshani@icd.uni-stuttgart.de, daniel.hernandez@ki.uni-stuttgart.de, steffen.staab@ipvs.uni-stuttgart.de, thomas.wortmann@icd.uni-stuttgart.de

Abstract: The Semantic Digital Twins Lab, held in winter 2023/24 at the University of Stuttgart, introduced computer science students to knowledge representation in the building industry. This paper highlights key insights and challenges, particularly in querying semantic and geometric data in building graphs. Students struggled with complex data structures and global coordinate computation—often beyond SPARQL's capabilities—but gained valuable experience and problem-solving skills. However, despite these challenges, the lab experience offered valuable learning opportunities and fostered innovative problem-solving skills among the students.

Keywords: Teaching, architectural modeling, geometric information, knowledge representation, semantic web technologies, IFC, logical inferences



DOI: 10.18154/RWTH-CONV-254895. Published in the conference proceedings of the 36. Forum Bauinformatik 2025, Aachen, Germany, © 2025 The copyright for this article lies with the authors. This publication, except for quotations and otherwise indicated parts, is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license.

1 Introduction

The design, construction, and operation of a building comprise multiple disciplines and processes which must be performed simultaneously during the lifecycle of a building [1]. Giving digital objects semantics allows integrating knowledge across different disciplines, and ensuring a correct alignment. A comprehensive review of the application of semantic web technologies in BIM highlights both the potential and current limitations of these approaches in AEC contexts [2]. Recent work has demonstrated how semantic technologies can support co-design workflows and enable cross-domain querying and reasoning in AEC settings [3], reinforcing the educational value of introducing these concepts through practical lab environments. A research project of the Cluster of Excellence Integrative Computational Design and Construction for Architecture (IntCDC) at the University of Stuttgart, tackles these challenges. This research project is a joint effort between two institutes, the Department of Analytic Computing, Institute for Artificial Intelligence (KI/AC) and the Institute for Computational Design and Construction, Department for Computing in Architecture (ICD/CA). KI/AC teaches a course on Knowledge Graphs, which presents the fundamentals of the Semantic Web. Although this course

combines lectures and exercises, it does not show their application in a particular use case or industry. At the same time, ICD/CA is involved in teaching the Building Information Modelling (BIM) seminar, which teaches several tools and practices of the building industry for creating, collaborating and exchanging information, but does not cover semantic web technologies in-depth. We thus designed a course where students practice Semantic Web technologies in the building industry. This new course was conducted during the winter semester of 2023/24. The course included lectures and tasks covering semantic web technologies [4] and knowledge representation pertinent to the building sector [5], [6], [7]. Ten students from diverse Master's programs—including Computer Science, Infotech, Informatics, and Autonomous Systems—took part in the course. We utilized the open, international standard (ISO 16739-1:2018) data schema that supports the exchange of digital data among heterogeneous applications, the Industry Foundation Classes (IFC) [8]. Students used several IFC files following the IFC data schema, including the duplex house (including several disciplinary representations) and structural models in an IFC format from the Semantic Enrichment challenge of the LDAC2022 Challenge¹ [9]. In the world of semantic web technologies, students mainly used the Resource Description Framework (RDF), the Web Ontology Language (OWL) [10], and the query language SPARQL. They worked in groups, they undertook tasks ranging from converting IFC files to RDF (LBD ontologies, ifcOWL [11], and developing their own parser), and visualizing 3D models and RDF graphs to integrating features into a web application with data loading and querying capabilities, leveraging GraphDB². Weekly sessions focused on group work, progress updates, and applying newly learned skills. The course concluded with tasks involving logical inference from semantic and geometric data, such as detecting pipe intersections and identifying structural supports. This paper examines the use of semantic web technologies in architectural modeling education, focusing on BIM data integration, knowledge graphs, and linking data from different disciplines.

2 Lab Structure

The Semantic Digital Twins Lab was organized into weekly tasks and a final comprehensive task involving semantic and geometric querying on graph-based building models. The lab aimed to introduce students to semantic web technologies and knowledge representation in the building industry. Learning objectives included mastering RDF conversion, querying geometric data, applying ontologies, and solving problems related to building information. Through hands-on and collaborative exercises, students gained experience in representing and managing architectural data using semantic web principles, with a focus on real-world applicability. Each week, students worked on tasks aligned with specific Intended Learning Outcomes (ILOs), covering skills such as defining software architecture, visualizing graphs, converting IFC files to RDF, using ontologies like ifcOWL and LBD, integrating systems, and building web interfaces. Students used different software, including the Java implementation of the Ruby language (JRuby), the Ruby RDF library, the Python RDF library RDFlib, the Apache Jena library for RDF file parsing, the GraphDB database for loading and querying RDF data with SPARQL, the Java tool for IFC to RDF conversion, the IfcOpenShell tool to facilitate IFC file reading and extracting relevant information, the D3 JavaScript library to visualize RDF graphs, and

¹<https://github.com/SSoLDAC-2022/Challenge-4>

²GraphDB is a graph database and knowledge discovery tool compliant with RDF and SPARQL

Protégé for ontology exploration. Table 1 outlines these ILOs and their corresponding tasks. The final task (ILO#14) required each group to tackle a complex problem in building industry knowledge representation. This involved integrating data across disciplines, exploring building elements, interpreting geometry, and understanding implicit vs. explicit information in models. Students deepened their proficiency in SPARQL, RDF, ontologies, and geometric representation. The following subsections present the four core tasks tackled by student groups, highlighting both outcomes and challenges.

Table 1: Intended Learning Outcomes (ILOs) of the Lab and Corresponding Tasks

ILO	Definition	Implementation Task
ILO #1	Develop the ability to define and conceptualize software architecture within the context of building industry knowledge representation.	Exploratory task for each group: define a software architecture.
ILO #2	Acquire the skills to integrate software components and systems within the context of building industry knowledge representation.	Software integration.
ILO #3	Improve graph understanding through visualization and interpreting building model representations.	Visualize graph.
ILO #4	Gain proficiency in converting IFC files to RDF format.	Convert IFC files to RDF graphs that do not follow a certain ontology.
ILO #5	Develop the ability to create a web interface for uploading IFC files, a crucial aspect of data management in the building industry.	Create a webform where users can upload IFC files.
ILO #6	Understand building models through geometry, thereby acquiring the skills necessary to visualize and interpret simple geometry on web platforms.	Visualize IFC geometry on the web (example geometry: Box).
ILO #7	Gain proficiency in parsing IFC files into existing ontologies, such as the Linked Building Data (LBD) ontologies.	Convert IFC files into LBD ontologies.
ILO #8	Develop the capability to store IFC and RDF files in a Graph Database, essential for data management and querying in the building industry.	Pushing the IFC and RDF files to Graph DB.
ILO #9	Understand building models through geometry, thereby acquiring the skills necessary to visualize and interpret IFC geometry on web platforms.	Visualize IFC file.
ILO #10	Gain proficiency in parsing IFC files into existing ontologies, such as ifcOWL formats.	Convert IFC files into ifcOWL graphs.
ILO #11	Develop the ability to define axioms within the context of building industry knowledge representation.	Defining Axioms, Exploratory task.
ILO #12	Acquire skills in inferring axioms, valuable for logical inferences in the building industry.	Inferring Axioms.
ILO #13	Gain proficiency in inferring facts and validating relationships using ontology editor software.	Infer facts using Protégé.
ILO #14	Learn data integration with semantic web technologies, by combining two disciplinary models for the same building.	Adding supports relation between architectural elements; adding connectsTo relation between transit spaces; linking pipes with architectural elements; linking electrical appliances with spaces.

2.1 Task 1: Adding “Supports” Relation between Architectural Elements

Task description: This task focused on inferring structural support relationships between architectural elements in the building model. Since these relations were not explicitly included in the IFC-derived ifcOWL data, students had to identify which elements likely support others—such as columns supporting beams—and represent this connection using a new predicate, “supports”. As the information was missing from the model, students were required to make assumptions and define heuristics to infer the relation. After adding this relation to the mode, the students must answer the following questions: (1) Which elements are not supported? (2) Which elements are supported by a specific element? (2) Which classes include elements that support other elements? (3) Which classes include elements that do not support other elements? **Students’ Solution:** The students assumed that the “support” relation can be inferred from the position of the building elements. If a building element is directly above another building element, then the students assumed that the element below is supporting

the element above. To simplify the operation with the irregular geometries of objects, the students assumed that this relationship can be inferred from the bounding boxes of elements. To reduce the search space, the students assumed that only certain types of building elements are designed to support other elements, the elements that have the attribute load bearing with the value true. They calculate the bounding box by finding the maximum and minimum values for the element coordinates regarding a global reference system. Hence, they translated these coordinates to a global coordinate system before computing the bounding boxes. They did this recursively for elements whose position is relative to a position of another element. Finally, they added the relation defined according to these assumptions, and they formulated the requested queries using SPARQL.

Evaluation of the learning objectives: Although the students completed the task, the goal of practicing with the fundamental concepts of ontologies was not met. In particular, they do not use the notion of inheritance of classes. Instead of finding arbitrary elements which support other elements, they find elements of particular classes (e.g., Column). Instead of exploring the ontology at a high-level, they explore patterns in the relationships between individuals of lower-level classes. Besides the planned objectives, they learned about the representation of coordinated systems.

2.2 Task 2: Adding “ConnectsTo” Relation between Spaces

Task Description: This task involved inferring spatial connectivity between rooms in the building model. While the IFC-derived data defined enclosed spaces, it did not explicitly state whether movement was possible between them. Students were asked to add this missing information by introducing a symmetric predicate, "connectsTo", between spaces that are directly connected. This connection could be inferred from the spatial geometry—either through openings like doors or direct adjacency without intervening elements. After adding this relation, students had to test if the relation was successfully incorporated by querying the SPARQL endpoint with the following queries: (1) Do we need to cross space A when transiting from space B to space C?, (2) Which spaces can be reached from space A without passing through another space?. **Students’ Solution:** The students’ approach was to utilize IFC opening elements and identify spaces related to the same ifcOpeningElement predicate. Figure 1 shows the steps the group took to tackle the task.

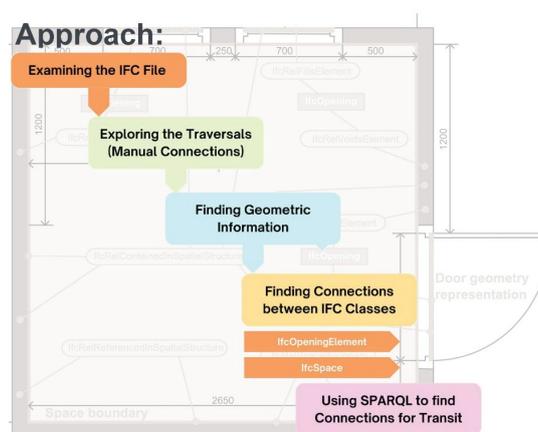


Figure 1: Approach of students in adding “ConnectsTo” Relation between Spaces.

Evaluation of Learning Objectives: Students completed the task by adding the "connectsTo" relation between some spaces. However, they did not succeed in correctly adding this relation, because they added the relation between spaces that are connected by an `ifcOpeningElement`. An `ifcOpeningElement`, in essence, can also be a non-transit element, such as a window. Nevertheless, they gained insights into spatial relationships and navigation pathways by analyzing the hierarchy of the ontology classes, as well as the physical representation of the building.

2.3 Task 3: Linking Pipes with Architectural Elements

Task Description: This task required students to integrate separate disciplinary models by inferring spatial relationships between pipes and architectural elements. Specifically, they had to identify where pipes intersect with building components, such as walls, and make this relation explicit using the predicate `crosses`. The intersection information had to be derived from the geometric data provided in the mechanical, plumbing, and architectural models. Finally, students were asked to test the success of incorporating this relation by answering the following queries: (1) Does any pipe cross a load-bearing wall? (2) Which walls are crossed by pipes? **Students' Solution:** Despite encountering two primary challenges—understanding the class hierarchy of IFC files and extracting global coordinates of objects—the student group successfully addressed the task. They manually explored the graph and utilized GraphDB to deepen their understanding of the data structures involved. Employing `IfcOpenShell` facilitated the retrieval of relevant information from the IFC, allowing them to insert it into the graph using SPARQL `INSERT DATA` queries. Subsequently, they crafted queries to retrieve coordinates of walls and pipes, followed by calculations of bounding boxes and intersections using SPARQL queries (see Algorithm 1, Algorithm 2, and Algorithm 3).

Evaluation of Learning Objectives: While successfully linking pipes with architectural elements, students engaged in manual exploration and utilized tools like GraphDB and `IfcOpenShell` to overcome challenges. This hands-on approach fostered a deeper understanding of data structures and practical application of SPARQL queries.

SPARQL Algorithm: Coordinates of Wall Geometry from IFC File

Algorithm 1: Query to retrieve the coordinates of walls (e.g., from an IFC architecture file).

```
PREFIX ifc: <http://standards.buildingsmart.org/IFC/DEV/IFC2x3/TC1/OWL#>
PREFIX wall: <http://linkedbuildingdata.net/ifc/resources20240127_210211/IfcWallStandardCase_>
PREFIX list: <https://w3id.org/list#>
PREFIX express: <https://w3id.org/express#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT DISTINCT ?wall ?Polyline ?CartesianPointList ?Point ?Point2 ?Point3 ?Point4 ?Point5
FROM <https://ex.org/duplex_pieter>
WHERE {
  ?wall a ifc:IfcWallStandardCase; ifc:representation_IfcProduct ?ProductShape .
  ?ProductShape ifc:representations_IfcProductRepresentation ?RepresentationList .
  ?RepresentationList list:hasContents ?ShapeRepresentation .
  ?ShapeRepresentation ifc:items_IfcRepresentation ?Polyline .
  ?Polyline ifc:points_IfcPolyline ?CartesianPointList .

  ?CartesianPointList list:hasContents ?Point ;
    list:hasNext/list:hasContents ?Point2 ;
    list:hasNext/list:hasNext/list:hasContents ?Point3 ;
    list:hasNext/list:hasNext/list:hasNext/list:hasContents ?Point4 ;
    list:hasNext/list:hasNext/list:hasNext/list:hasNext ?Point5 .
}
```

SPARQL Algorithm: Calculating Bounding Box for Pipes

Algorithm 2: Calculating pipe bounding box based on position, radius, and depth.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ifc: <http://www.buildingsmart-tech.org/ifcOWL/IFC4#>
PREFIX ex: <http://example.org/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?wall
WHERE {
  GRAPH <https://ex.org/new_pipe_data> {
    ?pipe a ex:IcfFlowSegment;   ex:x ?pipeX; ex:y ?pipeY; ex:z ?pipeZ;   ex:radius ?radius; ex:depth ?pipeDepth .
    BIND (xsd:double(?pipeX) - xsd:double(?radius) AS ?pipeMinX)
    BIND (xsd:double(?pipeY) - xsd:double(?radius) AS ?pipeMinY)
    BIND (xsd:double(?pipeZ) AS ?pipeMinZ)
    BIND (xsd:double(?pipeX) + xsd:double(?radius) AS ?pipeMaxX)
    BIND (xsd:double(?pipeY) + xsd:double(?radius) AS ?pipeMaxY)
    BIND (xsd:double(?pipeZ) + xsd:double(?pipeDepth) AS ?pipeMaxZ)
    FILTER (
      (?wallMaxX >= ?pipeMinX && ?wallMinX <= ?pipeMaxX) ||
      (?wallMaxY >= ?pipeMinY && ?wallMinY <= ?pipeMaxY) ||
      (?wallMaxZ >= ?pipeMinZ && ?wallMinZ <= ?pipeMaxZ)
    )
  }
}

```

SPARQL Algorithm: Inserting Intersection Data and Querying Crossings

Algorithm 3: Inserting new intersection triples and querying out :crossesWith relations.

```

PREFIX ifc: <http://standards.buildingsmart.org/IFC/DEV/IFC2X3/TC1/OWL#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX inst: <http://linkedbuildingdata.net/ifc/resources20231206_102530/>

INSERT {
  GRAPH <https://ex.org/intersections> {
    inst:Pipe_54976 ifc:crossesWith inst:Wall_4131 ;
                    ifc:crossesWith inst:Wall_4287 ;
                    ifc:crossesWith inst:Wall_4399 ;
                    ifc:crossesWith inst:Wall_4465 ;
                    ifc:crossesWith inst:Wall_5448 .
  }
}
SELECT ?wall
WHERE {
  ?pipe rdf:type ifc:IcfFlowSegment;   ifc:crossesWith ?wall.
  ?wall rdf:type ifc:IFCWallStandardCase .
}

```

2.4 Task 4: Linking Electrical Appliances with Spaces

Task Description: This task involved integrating electrical and architectural models by inferring the spatial location of electrical appliances within the building. Students were asked to establish explicit connections between appliances and the spaces they occupy using the predicate `locatedIn`. Since this spatial relation was not directly available in the IFC data, it had to be inferred from the geometric information. To validate the integration, students queried the followings: (1) Which room has the most electrical appliances?, (2) Which two appliances are very close to each other? (so that we can use the same circuit). **Students' Solution:** Students initially visualized the IFC files to comprehend their structure. They attempted to establish a linkage between the files but encountered challenges in aligning spaces. Additionally, they implemented functionality to find the closest appliances, but faced difficulties with SPARQL while integrating the electrical file with the architectural file due to disparities in space identification and the unavailability of global coordinates for the architectural file. These obstacles hindered the integration process.

Evaluation of Learning Objectives: Despite completing the task for some elements, the students faced challenges in determining coordinates for all rooms. Specifically, they struggled with computing the global coordinates of geometric representations due to the intricate nature of local placements. However, through these challenges, they gained a deeper understanding of building models. They still managed to implement the process of integrating two IFC files and delved into understanding the coordinate system, both relative and global.

3 Conclusion

The Semantic Digital Twins Lab introduced students to knowledge representation challenges within the building industry through practical, interdisciplinary tasks. The course offered hands-on experience with structured data modeling, querying, and inference in the context of digital building representations. The most significant challenges arose from understanding how building models are encoded using ifcOWL and from extracting topological and spatial relationships between building components. These geometries are often represented in local coordinate systems, requiring recursive transformations to derive global positions. Students found this process particularly demanding, as SPARQL offers only limited recursion capabilities and lacks support for non-commutative transformations such as rotations. We observed that these technical hurdles risk overshadowing the broader learning objectives of semantic modeling. To address this, future iterations of the lab will place greater emphasis on modeling fundamentals from the outset. Specifically, we plan to (i) enrich the current ifcOWL datasets with concepts from the BOT ontology to improve semantic clarity and (ii) introduce geometric encoding and coordinate transformation principles earlier in the course. The course was attended primarily by students from computer science and related disciplines. While they demonstrated strong proficiency in programming, RDF modeling, and database querying, difficulties with spatial reasoning and geometric interpretation highlighted the absence of architectural perspectives. In the future, we aim to include students from architectural and design backgrounds to better support interdisciplinary collaboration and domain-relevant problem-solving. Students responded positively to the structure of the lab. Working in groups fostered collaboration and knowledge exchange, while the interactive format—with regular feedback and integration between lectures and hands-on tasks—was well received. The diversity of tutor backgrounds, spanning computer science, AI, and architecture, further enriched the learning experience and was noted as beneficial in student feedback. These findings support our conviction that an interactive, interdisciplinary setup is essential for effectively teaching the application of semantic web technologies in the AEC domain. Overall, the lab demonstrated both the potential and the current limitations of integrating semantic web technologies in AEC education, offering valuable lessons for curriculum development at the intersection of BIM and knowledge representation.

4 Acknowledgements

Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2120/1 – 390831618 and COFFEE - STA 572_15-2. The authors thank all course participants: Ankita Saha, Annie Bhalla, Chunzhi Guo, Haomin Shi, Long Bai, Mayank Jha, Robin Schalkowsky, Shivani Soni, Varun Bhat, and Yao Chu.

References

- [1] D. Elshani, T. Wortmann, and S. Staab, “Towards Better Co-Design with Disciplinary Ontologies: Review and Evaluation of Data Interoperability in the AEC Industry”, in *Proceedings of the 10th Linked Data in Architecture and Construction Workshop (LDAC 2022)*, vol. Vol-3213, Hersonissos, Greece: CEUR Workshop Proceedings, 2022, pp. 43–52.
- [2] X. Zhang, J. Beetz, and M. Weise, “A Review of the Application of Semantic Web Technologies in Building Information Modeling”, *Archives of Computational Methods in Engineering*, vol. 29, pp. 1991–2022, 2022. DOI: 10.1007/s11831-021-09595-6
- [3] D. Elshani, A. Lombardi, D. Hernandez, S. Staab, A. Fisher, and T. Wortmann, “AEC Co-Design Workflow for Cross-Domain Querying and Reasoning Using Semantic Web Technologies”, *Automation in Construction*, vol. 176, p. 106 226, 2025. DOI: 10.1016/j.autcon.2025.106226
- [4] A. Hogan et al., *Knowledge Graphs*. New York, NY, USA: Association for Computing Machinery, Jul. 2021, vol. Vol-54. DOI: 10.1145/3447772
- [5] D. Elshani et al., “Building Information Validation and Reasoning Using Semantic Web Technologies”, en, in *Computer-Aided Architectural Design. INTERCONNECTIONS: Co-Computing Beyond Boundaries*, vol. 1819, Cham: Springer Nature Switzerland, 2023, pp. 470–484. DOI: 10.1007/978-3-031-37189-9_31
- [6] P. Pauwels and K. McGlinn, editors, *Buildings and Semantics: Data Models and Web Technologies for the Built Environment*, First edition. Leiden, The Netherlands ; Boca Raton, FL: CRC Press/Balkema, 2023. DOI: 10.1201/9781003204381
- [7] D. Elshani, A. Lombardi, A. Fisher, D. Hernandez, S. Staab, and T. Wortmann, “Knowledge Graphs for Multidisciplinary Co-Design: Introducing RDF to BHoM”, in *Proceedings of the 10th Linked Data in Architecture and Construction Workshop (LDAC 2022)*, vol. Vol-3213, Hersonissos, Greece: CEUR Workshop Proceedings, 2022, pp. 32–42. [Online]. Available: <http://ceur-ws.org/Vol-3213/>
- [8] buildingSMART International, *Industry Foundation Classes (IFC)*, buildingSMART International, 2018. Accessed: Oct. 10, 2024. [Online]. Available: <https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/>
- [9] T. Bloch and R. Sacks, “Clustering Information Types for Semantic Enrichment of Building Information Models to Support Automated Code Compliance Checking”, *Journal of Computing in Civil Engineering*, vol. 34, no. 6, p. 04 020 040, 2020. DOI: 10.1061/(ASCE)CP.1943-5487.0000922
- [10] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, *OWL 2 Web Ontology Language Primer (Second Edition)*. W3C Recommendation, <https://www.w3.org/TR/owl2-primer/>.
- [11] P. Pauwels and W. Terkaj, “EXPRESS to OWL for Construction Industry: Towards a Recommendable and Usable ifcOWL Ontology”, *Automation in Construction*, vol. 63, pp. 100–133, 2016. DOI: <https://doi.org/10.1016/j.autcon.2015.12.003>