Semantic Web Ontologies for Building Objects and their Performance Data

Eleanna Panagoulia, Zachary Lancaster and Tarek Rakha
School of Architecture, Georgia Institute of Technology, Atlanta, GA, USA

Abstract

While the Architecture Engineering Construction and Owner-operated (AECO) industry has been successful in digitizing data concerning buildings through Building Information Modeling (BIM) applications, transforming these data into usable digital services (digitalizing) has not been fully addressed. The Semantic Web allows for the creation of abstraction layers that enable building data as a service. This paper proposes Semantic Web ontologies for representing buildings, the relationships between their elements and analytical data, along with attendant annotation systems. This method enables bi-directional exchanges between heterogeneous platforms, introducing flexibility in representing, sharing and re-using data. The work demonstrates a framework for the digitalization of building data and a service-oriented model, improving stakeholder collaboration.

Key Innovations

- A framework of three vocabularies for Linked Building Data (LBD) concerning building object geometry and performance analytics data.
- An annotation system to move between design and analytical software and a graph data store.
- A means of handling unstructured, non-native data in architectural projects.

Practical Implications

This paper demonstrates a practical example of ways to implement Semantic Web concepts into practice in a real-world context. This provides architects and engineers with a basis upon to build further knowledge on applying these methodologies in their own work and make use of both the proposed and other existing LBD vocabularies.

Introduction

The paradigm of specialist centric practice is rapidly giving way to more interactive processes meant to provide design practitioners a more intuitive understanding of the evaluation of building performance (Zanni et al., 2017). The specialist mode of working within the AECO industry has been necessary due to the persistent problem of a lack of accurate and accessible means of exchanging data between software platforms. This requires specialist knowledge to translate design models into compatible representations for Building Performance Simulation (BPS) and to calibrate analytical models manually (Negendahl, 2015), (Gao, Koch, and Wu, 2019). The outcomes of these models become, in essence, trapped within their bespoke, analytical environment and are functionally inaccessible. Moreover, there are additional challenges regarding the combination of data from diverse resources, which necessitates extensive human labor given the diversity of the large amounts of data incorporated in the AECO industry. Although BIM provides a foundation for collaboration, convergence of different forms of data and data exchange within a common platform, professional practice still relies heavily on document sharing to circumvent the limits of proprietary software ontologies, especially when integrating performance analytics (Farzaneh, Monfet, and Forgues, 2019). Current approaches of using relational data schemas to communicate information (i.e., Industry Foundation Class (IFC)) constrain the domain of possible content, hence reducing the data expressivity and failing to capture the diversity of data in the built environment (Bonduel et al. 2020). BIM plays a central role in contemporary design and construction practice, serving in theory, as a database of record (Azhar, 2011). The inability to communicate results directly into BIM hinders the applicability of building performance data, as the exchange between BPS tools and BIM does not meet user expectations on exchange, conversion and concurrence. We note that even if the data exist in machine-readable format, users could encounter difficulties in finding and analyzing them without a data annotation system in place, to ensure that a common concept is not being interpreted in different ways by different users. This lack of semantic interoperability is exacerbated in the context of existing buildings particularly for energy auditing and diagnostics, as data are often unstructured and in formats that are non-native to the bounded representational spaces of proprietary software (Kamel and Memari, 2019).

BIM has been successful in the process of digitizing data concerning buildings, but remains a passive repository. The digital transformation of BIM into a queryable building Data as a Service (DaaS) platform, cannot be achieved through the current set of monolithic schemas (i.e., IFC), as they require wholesale model translation for exchange, instead of serializing only relevant model data (Chen, Jin, and Alam 2018). There is recognition of the need for BIM software to surpass geometry representation and move towards a service-oriented data store capable of combining, linking, querying and reusing data concerning buildings (Sattler, Lamouri, Pellerin, and Maigne, 2019). This database model should represent geometry, but should also store and retrieve analytical and operational data (Gilani, Quinn, & McArthur, 2020).

Advances in Information and Communication Technology (ICT) have led the development of new information systems that contextualize data in a way that facilitates exchange and cross-domain interoperability between software, but also allows for schema free representations. The Semantic Web is a means of publishing domain specific knowledge and relationships, in the form of semantic ontologies, which play a role in linking data between different semantic contexts (Hinze et al. 2012). Ontologies are vocabularies that describe objects (classes) and relationships (properties) within a knowledge domain in the form of triples (subject, predicate, object) (Gruber, 1993). This allows linking machine-readable data about a knowledge area with its counterpart in another, via semantic web Universal Resource Identifiers (URI); a critical step in data exchanges in software systems. These technologies have the potential to address the persistent limitations of interoperability between BPS tools and design software, as well as manage effectively the heterogeneity and large volumes of data in these domains (Corry et al., 2015). Semantics aid the digitalization of building data and the conversion of BIM from a passive repository towards a building-oriented data service, by providing reusable, extensible, and machine-readable data webs.

The purpose of this work is to demonstrate a method for managing building data, including geometry and performance analytics, into a queryable form, as well as to demonstrate the benefits semantic web ontologies have in maintaining data integrity between, otherwise, disjoint environments. This paper is organized as follows: First, a brief overview of existing semantic web ontologies within the AECO knowledge domains, identifying the gap the proposed ontologies cover. Second, the proposal of a framework of three novel vocabularies that address the specific semantic needs of linking building objects with their related performance data. The first vocabulary, Ontology for Building Objects (OBO), describes building objects and their geometry in a consistent alignment to BIM. The second, Ontology for Building Performance Data (OBPD), describes performance input data and properties necessary for the operation of BPS analytics (e.g., material properties) and serves as a link between the first and the third. The third, Ontology for Performance Analysis (OPA), describes performance analyses configuration, output data and performance objectives, including energy audits. Third, a method for linking BIM and analytical data via an annotation system that allows data serialization from BIM to an analytical environment, storage and retrieval via a knowledge graph. Lastly, this paper demonstrates a vertical slice of this method.

Semantic Web Applications within AECO

The lack of semantic interoperability and standardization on data representation, has been a persistent issue within AECO and has limited the ability to communicate information between stakeholders. Although semantic web technologies present benefits for the above challenges, and there is a clear interest in how data concerning buildings are communicated, it is important for the building community to understand the scope and overlap of existing development. In specific, for ontology development, building data are categorized as either static or dynamic, where static data types describe physical elements, while dynamic types describe time series data (i.e., for building operations). According to Gelani, 90% of existing ontologies include both static and dynamic data types, while 10% include only static data types (Gilani, Quinn, & McArthur, 2020). Some of the most popular ontologies associated with dynamic data types, such as smart building operations and smart device integration, are BRICK and SAREF (BrickSchema, 2016; ETSI, 2013). Examples, of ontologies associated with static data types are BOT (Rasmussen, 2021), which describes topological relationships between building components and is aligned with building operations ontologies, and ifcOWL (Terkaj & Pauwels, 2017), which represents the IFC schema for building construction data in the web ontology language (OWL) (OWL, 2012). A limited number of ontologies focus on performance analytics, such as SimModelRDF (Pauwels, Corry, O'Donnell, 2014), which is an OWL translation of the SimModel XML schema and Performance Framework (PF), a high-level description of the relations between performance metrics and performance objectives (Corry et al. 2015). Other instances of ontologies are related to building heritage and preservation for documentation purposes (Pauwels, Zhang and Lee, 2017), as well as life cycle management and environmental assessment at the product level for buildings (Tchouanguem et al., 2019).

A review of existing ontologies, by Pritoni et al., identifies gaps and limitations in representing buildings and performance aspects (Pritoni et al., 2021). The majority of the limitations of current ontologies relate to linking physical building objects with data and properties. Ontologies, such as BOT, describe building topologies at a high-level, but not specific parts of a building (i.e., envelope, window). Building management ontologies, such as BRICK and SAREF, are too specific around low-level information, such as sensors and devices, without linking these data to spaces and geometric components of the building adequately. Rather, these ontologies structure existing metadata concerning sensors and systems already present in building management systems (Bergmann et al., 2020). Moreover, some concepts are not currently well represented, such as building envelopes, energy audit data and analysis outputs links to objects. The proposed ontologies attempt to cover the above limitations.

OBO describes building objects, however, it addresses the usability and over-specification issues of ontologies, such as ifcOWL, which lacks hierarchical structure that challenges implementation. This issue has been previously identified by Tejkaj , as well as a need for a new ontology to cover formal building object information (Terkaj & Pauwels, 2017). ifcOWL is, while a robust representation, closely aligned to, and incapable of capturing concepts not present in IFC. OBPD provides a vocabulary for describing object properties related to performance (i.e., Energy Plus materials) and captures many of the same concepts as SimModelRDF, while introducing new ones, such as the ability to link external documents to objects. Although SimModelRDF contains many of the concepts, its limitations are the same as in the ifcOWL case in terms of structure and direct alignment with SimModel (Pauwels, Corry, O'Donnell, 2014). OPA addresses the limitations of linking performance objectives with analyses outputs, as well as linking these outputs with the relevant object geometry. While SimModelRDF links properties to analyses outputs (input/output) and PF (Corry et al., 2015) links objectives to metrics, the inter linking of properties with analysis outputs and performance objectives is still not possible. Moreover, OPA captures the concept of energy audit data. While SimModelRDF provides input fields for properties, it does not provide a meta-description of the required parameters for each analysis type. The proposed ontologies create a foundation for additional ontology research to create richer descriptions of buildings and their linked data (Bergmann et al., 2020).

Proposed Ontology Framework

To describe buildings and their performance data as a knowledge graph, we have identified three distinct vocabularies, OBO, OBPD and OPA that support the ways that BPS data are used and understood in relation to BIM models. We propose a federated knowledge model to handle these vocabularies, which takes advantage of the OWL open world assumption, where each conceptual domain has its own definition and is aligned with others to form a larger framework (OWL, 2012). The division of the vocabularies makes each knowledge domain more comprehensible, as each contains only information necessary to encapsulate a single layer of knowledge. We focus our ontology design starting point on geometry objects, as the principal data container, and attempt to map information deriving from BPS to those objects, either at a high-level, for instance, a zone or at a lower level, for instance, individual elements, such as walls. This design assumes that, within AECO, any element that can hold data will have representational geometry. The main classes of the three ontologies and alignments with other ontologies can be viewed in Figure 3 at the end of the manuscript or via a link to GitHub that can be found below. https://github.com/PerformanceInformatics.

Ontology for Building Objects (OBO)

OBO provides a stable vocabulary common to architectural practice for representing building objects and linking data to them (Azhar, 2011). It serves as a bridge between Computer Aided Design (CAD), BIM software and a knowledge graph. OBO enables the enrichment of CAD/BIM data with metadata, such as performance analytics. ISO 12006-2:2015 presents a framework for building construction objects and serves as a basis for OBO's understanding of building objects. It provides an explicit and consistent language for describing elements that make up a building and the relationships between these objects. OBO consists of seven top level classes with seven primary subclasses that represent specific concepts, and thirty-two subclasses beneath those representing specific cases. From the seven top level OBO classes, five classes represent conceptual constructs, while the other two represent physical building objects. Four of the conceptual constructs are classes, Project, Site, Building and Space, that represent different levels at which aggregations of building objects may be understood. These aggregations of elements aid in the alignment of OBO to other ontologies, such as BOT and ifcOWL. These classes allow elements to be retrieved at higher levels of aggregation for easier manipulation (e.g., all walls related to a specific space) during implementation. The fifth conceptual class, Collection, represents discrete collections of elements and its subclasses are Interior, and Envelope. This organization aids in the identification of envelope and interior sets of geometry, and also the mapping of data related to BPS and envelope audits. The OBO classes related to physical objects are Element and Material. The class Element is a core concept to OBO and contains the subclasses representing the terminology for representing architectural objects, which cannot be further conceptually subdivided and are, therefore, semantically complete within themselves. The Element class has three subclasses, AbstractElement, AssemblyElement and UnitElement, representing different formal concepts of semantically complete architectural entities. UnitElement subclasses describe physical construction elements that cannot be further decomposed into other elements at a given level of detail and that make up larger construction systems. AssemblyElement represents a set of physical objects that are assembled with an ordered, tree-like hierarchy, where the relationship between objects is substantive and the removal of any object would compromise the semantic completeness of the assembly. Examples of AssemblyElement are curtain systems (class CurtainSystem). Curtain systems consist of mullion, transom, and panel elements in which the relationship is hierarchical (the mullion grid holds panels). It is clear in this example that if mullions were removed from the assembly, the remaining geometry could not itself be considered a curtain system. AssemblyElement is distinct from Collection due to the ordered and hierarchical nature of the relationships (AssemblyElement implies a tree topology, while Collection implies a star topology). It is possible to remove an element from a collection and not compromise the graph node hierarchy (ring, star or mesh node topology), while removing a hub node from an assembly element would immediately remove the nodes that depend on it hierarchically (tree-like topology with more than one levels of nesting). Figure 1 showcases different network topology instances of the Collection class versus an AssemblyElement class. Succinctly, a Collection is agnostic to the types of elements it contains, whereas an AssemblyElement is not. AbstractElement describes elements of non-physical or material substance, for instance, analytical representations of objects, such as surfaces and boundaries, topological interfaces between objects, as well as helper objects, such as annotations. The class, Interface, is an abstract concept to describe conceptual adjacencies between objects, for instance zones and rooms adjacencies in a building. The class Surface represents the analytical representation of an element for example, compatible for BEM simulation, while the class Boundary represents the two-dimensional boundary of certain objects (i.e., zones, floors) that enable processes, such as area calculation. Class Opening, describes areas where an otherwise continuous solid volume is disrupted. Lastly, Location is a means of describing the placement of an element within a logical coordinate system. The Annotation class represents concepts that denote ancillary data, such as level and grid, but also tags, text and URI references to other documents.

OBO Object properties, which describe relationships between classes, follow a similar architecturally based logic as the OBO classes, allowing for both real (material) and virtual (abstract) relationships. This is handled through two top-level properties, each describing a different relationship between objects and the semantic integrity of the relationship. First, materialRelation describes any relationship between any two elements that if absent, would compromise the integrity of the overall represented structure. Thus, a materialRelation describes relationships in building construction systems and these are related to either construction systems (i.e., madeOf). Moreover, physicalRelation (i.e., isPhysicallyAdjacent, joinedTo) describes the ways in which objects physically relate with one another, either by proximity (adjacency), intersection, geometric union (join) and arrangement in elevation (an element is on top/bottom of another element). On the contrary, abstractRelation, represents any relationship between two elements, where the relationship has no physical characteristics and if violated, both objects could continue to exist independent of one another. These relationships allow for the construction of larger objects or concepts depending on the nature of the relationship.

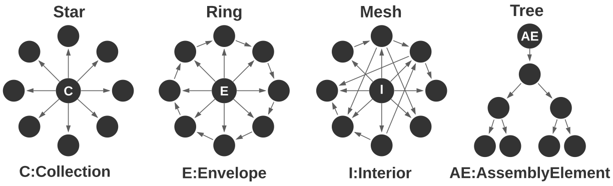


Figure 1: N*etwork topologies for OBO Collection, Envelope, Interior and AssemblyElement classes*

Ontology for Building Performance Data (OBPD)

OBPD is an ontology for representation and linking of performance related data in a consistent vocabulary common to BPS practice (ASHRAE, 2016). OBPD focuses on static data concerning properties of building objects, for example the thermal properties of a wall. Currently, OBPD consists of seven classes, and six subclasses that describe concepts that are directly related to performance data and six object properties that link domain-specific concepts to representational geometry. Relating performance data with geometry, through OBPD's alignment to OBO, we enable performance data to reside at an appropriate level of detail within a model, simplifying the process of conversion to a different epistemic context without data loss. OBPD's classes capture overarching BPS concepts that relate to building entities. The main concepts include the classes Zone, which aligns with OBO, for the description of a thermal zone, Schedule, the description of the level of operation of a system (i.e., equipment, lighting, HVAC) over a period of time, System, the description of a mechanical system for conditioning within a building and Program, the description of the type of activity taking place within a space. OBPD contains the class Document, which represents any external document that is related to an analysis, for instance, a set of infrared images deriving from an energy audit. The last and most elaborate class object in OBPD is the Material class, which represents the numerical properties of a single abstract material based on specific input data properties required for different analysis types. Currently, OBPD has subclasses for material types, such as EnergyPlusMaterial and RadianceMaterial, each with the relevant subclasses that represent various types (i.e., No Mass, Air Gap, Opaque etc.). These subclasses are defined by invoking the relevant data properties (i.e., density, roughness, R value etc.). The Material class does not store material thickness and assembly layer information. This is handled by the MaterialAssembly class, which describes the enumerated list of layers that make up an assembly and their thicknesses. MaterialAssembly consists of instances of its subclass Layer, following the principles of an Energy Plus construction. In this way, thickness and enumeration properties are assigned via the Layer class rather than the Material class. This allows for material types to be created, stored and reused independently to the specific layers of assembly used in a simulation that may be unique to a particular project. This method also maintains maximum flexibility as tools continue to develop over time as alignment to a different input schema becomes a matter of extension (adding new properties or classes) or serialization (finding common defined terminology).

OBPD has five object properties that allow the assignment of properties to physical OBO objects (i.e., hasMaterial, hasAnalyticalRepresentation). Although OBO contains properties for topological adjacencies, the same property, isAdjacent, is offered in OBPD, under a closed world assumption, since adjacencies between zones are relevant for simulation and are available independent of OBO. OBPD has six data properties that represent fields related to material thickness, enumeration and physical properties (i.e., Energy Plus, Radiance), program related properties, such type, systems schedules and rates (i.e., infiltration, flow rate etc.) setpoints for HVAC systems (i.e., cooling, heating, humidity setpoints etc.) and metrics regarding area and volume calculations.

Ontology for Performance Analysis (OPA)

OPA is an ontology for the description of performance analyses, as opposed to OBPD's representation of fixed, per element, performance data. OPA describes the overarching concepts of the objectives of an analysis, the type of analysis, its assumptions, requirements and outputs. OPA is separate from OBPD as they fundamentally describe two different aspects of the knowledge domains. While OPA provides a vocabulary to describe an analytical schema, OBPD provides a vocabulary to describe the properties, related to an

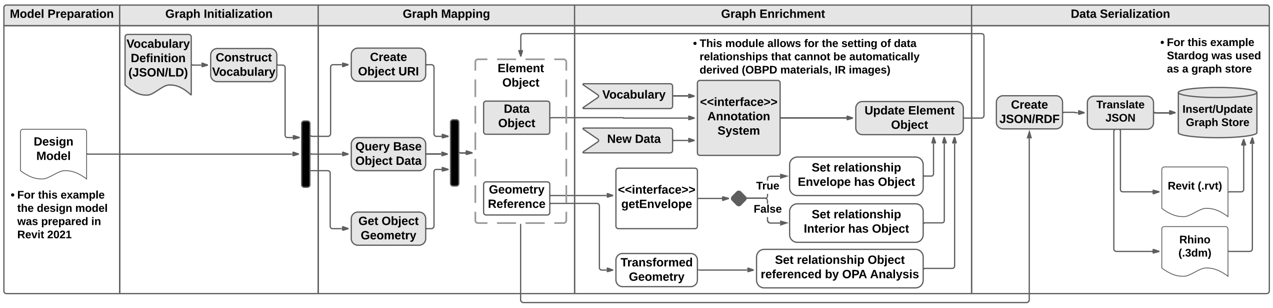

Model Preparation
Graph Initialization
Graph Mapping
Graph Enrichment
Data Serialization
Design Model
• For this example the design model was prepared in Revit 2021
Vocabulary Definition (JSON/LD)
Construct Vocabulary
Create Object URI
Query Base Object Data
Get Object Geometry
Element Object
Data Object
Geometry Reference
• This module allows for the setting of data relationships that cannot be automatically derived (OBPD materials, IR images)
Vocabulary
New Data
<<interface>> Annotation System
Update Element Object
<<interface>> getEnvelope
True
False
Set relationship Envelope has Object
Set relationship Interior has Object
Transformed Geometry
Set relationship Object referenced by OPA Analysis
• For this example Stardog was used as a graph store
Create JSON/RDF
Translate JSON
Insert/Update Graph Store
Revit (.rvt)
Rhino (.3dm)

Figure 2: Case Study Framework Diagram showing the main processes included in the five modules

analysis input of abstract (i.e., zone) and physical elements (i.e., wall). Keeping these as separate ontologies improves maintainability and extensibility.

OPA presents a vocabulary encapsulating common type of analyses relevant to BPS (Mahdavi and Wolosiuk 2019). OPA consists of six classes and ten subclasses. OPA currently provides four classes for describing categories of analyses, Energy, SpatialComfort, Daylight and EnergyAudit. These classes store the assumptions and input requirements necessary for a single run of a given type of analysis under a specific set of assumptions. This differs from OBPD which stores the actual input values for the various building object properties that relate to performance. An additional class, Evaluation, represents the output data derived from a single analysis. The PerformanceObjective class describes a performance goal and is related to any analysis type via the property requiresAnalysis. This offers the opportunity to relate performance targets with the analysis outcomes. OPA's nomenclature is flexible, general and high-level when compared to OBO and OBPD. OPA provides the nomenclature for describing fundamental analysis types, by providing its classes the necessary data settings from the data properties, inputSpecification and assumption. By defining OPA analysis objects to the necessary OBO elements, via OPA object property references, it is possible to fully represent all elements involved in an analysis and the necessary data for the creation of an analytical model and link the resulting data back to their host objects. The current state of development currently supports only the four broad types of analyses above. The open nature of OPA as a project provides a forum for the wider field to extend the range of descriptions related to analysis and performance in the future.

Knowledge Graphs and Annotation Systems

Data hold meaning only implicitly and within a specific knowledge domain or an a priori semantic context (such as a schema). BIM provides a robust, but rigid, semantic that serves as a connectivity map between disciplines that use this platform. This map also serves as a basis for understanding the link between geometry and building data (Pauwels, Zhang, and Lee, 2017). There is a need to advance BIM as a concept towards one that supports a broader array of data and model representations. Knowledge graphs, unlike relational databases, support flexible definitions and are well suited for heterogenous data found in the AECO industry and well suited for applications with high read to write ratios. The role of an ontology in this context, is to serve as the backbone structure upon which data would be organized in a knowledge graph data store. The knowledge graph operates as a back-end that manages model data, having a similar role to Revit's current database, with the only difference being that the graph would not directly store geometry files. Rather, the graph would store references to that geometry in other files (i.e., Revit models), since storing large binary objects creates technical problems for data management software. The proposed ontologies are a component for the knowledge graph implementation and a starting point in expanding the existing domain of BIM object representations towards a wider range of concepts for building objects, that can be queried based on the relevant knowledge domain of an objective.

To make full use of a knowledge graph-based BIM, users would need to access an application-level annotation system that provides a semantic context to objects in a more flexible way. These include automation for identifying, assigning and aligning semantic definitions as data move between environments. In the case study, the authors have developed several modules that, while operating in the Rhino/Grasshopper environment, present an agnostic means of achieving a portion of the necessary automation. The modules provide semantic alignments between OBO and Revit, as well as maintaining the object relationships between Revit objects and Revit objects converted to their analytical representation in Rhino (i.e., surface representation of Revit elements) automatically. For additional relationship establishment that exceeds geometry transformation alignments, the authors created a semi-manual annotation system, where the user establishes the subject-object relationship based on the domain of classes offered by the proposed ontologies (i.e., IR image mapping to the appropriate wall). These types of processes are necessary for any interoperable system whether that makes use of knowledge graphs or not.

Combined, a knowledge graph and an application, as the one described above, would form the basis for a digital transformation of the AECO industry and are key technologies for the next generation of BIM. The next generation of BIM tools should enable users to communicate concepts, data, and model representations presented in a form familiar to the knowledge domain of the user. This overcomes the difficulty of different interpretations of the same concept that create semantic incongruencies within a multidisciplinary environment.

Case Study

The case study presents a framework for interoperable processes between BIM, BEM and a knowledge graph that represents a queryable data source that is not beholden to any single software schema. The demonstrated workflow showcases a smooth process of sharing data between industry standard software, Autodesk Revit, McNeel Rhinoceros 3D (Rhino), Ladybug Tools, in Rhino/Grasshopper, as a front end to the Energy Plus simulation engine. The case study shows a vertical slice of ongoing development that focuses on the critical data handling portion of an interoperable system. The specifications for the slice are to make use of the above proposed ontologies to: (a) translate data from a BIM platform (Revit) into an Energy Plus BEM model, created in Ladybug Tools, (b) transmit these data from BIM and BEM to a queryable knowledge graph hosted in Stardog Cloud (Stardog, 2005) and (c) return output BEM data to the BIM environment, through a workflow developed in Python and Rhino/Grasshopper. The current implementation makes use of Revit's internal annotation system, which is the existing software schema for element categorization, as a way of formally mapping existing Revit element categories with additional properties that go beyond Revit's built-in parameter list, but do not exceed Revit's basic semantic. For example, this work showcases the integration of external data to a built-in Revit element by extending the domain of its parameters, but it does not involve the creation of new types of Revit element categories. A schematic representation of this process is presented in in Figure 2, and modules relevant to the vertical slice are highlighted with grey fill. The vertical slice is comprised of five modules, described below. These modules take advantage of object-oriented best practices to create a robust object model that maintains data consistency throughout the process in a tree-like form in, which each branch represents a singular object and semantic annotations are used to relate branches together. The subject for this vertical slice is a residential, three-zone, two-storey building built in 1950 and retrofitted within the last decade. The building was surveyed by the authors for vulnerabilities related to thermographic anomalies in the building façade.

The first module of the framework prepares the design model for ingestion into the process, in this case, a Revit model. Revit's robust existing semantics make extended automation possible. This module makes use of the Rhino.Inside.Revit plug in to provide simultaneous access to Revit and Rhino/Grasshopper environments, allowing the Revit elements and their data to be accessed. Current requirements for the framework execution include a Revit model that contains material assembly information, spaces and rooms to define zones for energy simulation and its elements should be organized in worksets.

The second module initializes the graph within Rhino/Grasshopper. This step builds the vocabularies for the three ontologies in JSON-LD format, which is exported from the ontology editor. The JSON-LD is parsed in ghPython due to its consistent structure of exposing the object classes, object properties and data properties of each ontology to the automation and annotation components of the following modules.

The third module is a type of annotation system, which maps BIM elements with the ontology vocabulary to create data objects that contain class type, a URI and identity properties in the design environment (i.e., RevitElementID), with boundary representation (brep) geometry interpreted by Rhino.Inside.Revit. This module maps the elements based on Revit's built-in semantics to those contained in OBO, for example a Revit Wall is equivalent to the OBO object Wall. This module has multiple sub-processes that take place as a part of enriching graph objects: (a) a query process of the Revit elements via Rhino.Inside.Revit, (b) a module that maps relevant Revit element types to valid OBO classes, (c) a process that assigns the valid OBO and OBPD classes to the data objects inside Rhino/Grasshopper.

The fourth module enriches the previously constructed semantic objects with new data not otherwise represented within their native context. This process has sub-modules that augment the objects with new data, in this case, IR images as part of an energy auditing process. These modules include automation for: (a) detecting the members of the BIM envelope topologically via examining the number of the max surface area faces that are adjacent to a zone (i.e., one adjacency means that the wall is part of the envelope), (b) transforming Revit geometry (windows, walls, zones etc.), imported in module one, into usable geometry for Energy Plus while maintaining their relationship to their originating data, (c) update objects, residing within memory, with data resulting from the energy model. This module enables a translation between Revit geometry and a commensurate zonal (single zone or multi-zone) energy model, complete with windows and other openings, such as doors. These data are transmitted to a knowledge graph via automatically generated RDF triples in either JSON-LD or N-Triples formats and loaded to the knowledge graph. The triples are inserted into the cloud-based, w3c compliant, knowledge graph platform Stardog. Since each object ID is linked to its Revit element ID in the BIM model, it is possible to add these data in the form of additional parameters in the BIM model, or query the data model, represented in the knowledge graph through the query language, SPARQL. Figure 4 showcases a conceptualized graph using the three ontologies to represent an example of one of the second-floor zones, an envelope wall, an interior wall, an IR image and energy loading output data derived from an energy simulation.

The fifth module is a data interpreter that serializes native and foreign data back to the original design environment. Data are obtained either by querying the constructed knowledge graph or using in situ semantic objects produced in earlier modules, that take advantage of the JSON structure. In this way data from different processes, such as energy simulation, are serialized and integrated in the BIM model by way of the knowledge graph (Figure 5). The current development of the vertical slice took advantage of objects created in Grasshopper/ghPython, as opposed to querying the knowledge graph for integrating additional data to the Revit model. This method showcases how general object properties, energy audit and simulation output data can be aligned to a design model using the ontologies introduced in this paper. The vertical slice demonstrates a small portion of a larger system that allows for more direct understanding of the relationship between elements, their condition and performance inside the design environment, allowing designers to use diverse data types in their workflows.

Discussion

The research discusses the opportunities for transforming prevailing models, in our case, data modeling in the built environment, such as the replacement of traditional, relational data stores with the Semantic Web. The latter enables access to information based on multiple vectors that produce multiple ‘readings’ of a building. A linked data paradigm is by nature modular, allowing for any vocabulary that fits specific purposes to be added, while maintaining the integrity of the core. Unlike schemas like IFC, that require singular major releases to add additional vocabulary, a linked data approach is agile and supports modular development akin to open-source models. We believe that the methods for storing and managing data should not impose constraints on the information content. Users should be able to structure data to support their needs, rather than trying to bend practice to conform to rigid data structures. A technical benefit of the Semantic Web is the ability to maintain parallel representations of models that can be re-serialized in a consistent and complete way (Rasmussen, Lefrançois, Pauwels, Hviid, and Karlshøj, 2019). This allows access to the universe of possible data within the project, but also to pivot the whole model around domain-specific representations.

In the context of the ontology development and the case study, it is argued that data about building performance must link back to formal building elements, as these elements are common components to both the design and performance analysis practice. In this way, the geometry data layer becomes the coherent substrate upon which other data layers can be overlaid. Treating geometry as the central unit of exchange aligns well with how buildings are understood in practice. Buildings are understood as objects, not as collections of abstract symbols (Stiny, 2006). We propose to imbue architectural elements, in the form of representational geometry with their data and alternative representations. This results in a semantically consistent model across all representations and allows a single model store to support multiple disciplinary readings of the building based on practitioner needs.

Limitations and Future Work

The main limitation of this work is the absence of quality control mechanisms that ensure the design model meets the requirements described in this method. As this work is at an early stage, future technical work will focus on the development of an extensible abstraction layer for AECO that links across heterogeneous domains of knowledge and further integration and automation of the annotation system. Each ontology requires continued development, especially OBPD, to capture a wider range of Energy Plus properties, such as expanded window properties, to better map to Energy Plus documentation, while avoiding over-specification, such as in ifcOWL or SimModel. Further development on the annotation system requires improving automation modules to support more cases. Additionally, a necessary piece of further development is quality control modules that check the graph for completeness.

Conclusion

This paper presented a description of three ontologies, and an annotation system for traversing between BIM and BEM environments through a graph data store. The above have been showcased in a case study that features the relevant concepts around data transformation and management introduced by the authors. The authors’ objectives were twofold; first the development of an ontology framework to address the existing limitation of representing a range of high-level to low-level building objects and linking their existing properties with additional ones that relate to external, diverse processes, such as performance simulation outputs and energy audit data and second; addressing missing concepts that are relevant to contemporary practice, such as building envelope representation, energy audit processes and connection of performance analytics, with objectives and the relevant building objects. The paper presented a framework for the digitalization of AECO practices on architectural design and performance analytics. It sets the foundation for an initial implementation of querying, transforming, and linking data between BIM design software and performance analytics, more specifically simulation and thermal inspection. It provides an initial method for serializing models and integrating diverse data layers to a metamodel. There is a need for a means of modeling data about buildings through a singular abstract software interface to aid in development of consistent interoperable applications. This interface calls for the agility to handle the plethora of data types and concepts present within real-world projects, while being able to provide a consistent data interface that allows for descriptions of a building at multiple levels. This interface enables a new generation of applications, which serve to use data as a means of collaboration between disciplines.

Acknowledgement

This material is based upon work supported by the U.S. Department of Energy’s Office of Energy Efficiency and Renewable Energy (EERE) under the Buildings Technology Office Award Number DE-EE0008680.

References

American Society of Heating, Refrigerating and Air-Conditioning Engineers (2016). *Energy Standard for Buildings Except Low Rise Residential Buildings (ASHRAE 90.1-2016 (I-P)).*

Autodesk. *Revit.* V.2021.1. Autodesk. Windows 10. 2020

Azhar, S. (2011). Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry. *Leadership and Management in Engineering* 11 (3), 241–52.

Bergmann, H., Mosiman, C., Saha, A., Haile, S., Livingood, W., Bushby, S., … Pritoni, M. (2020). Semantic Interoperability to Enable Smart, Grid-Interactive Efficient Buildings. *ACEEE Summer Study in Energy Efficiency in Buildings*, 44–59.

Bonduel, M., Wagner, A., Pauwels, P., Vergauwen, M.and Klein, R. (2019). Including Widespread Geometry Formats in Semantic Graphs Using RDF Literals. *Proceedings of the 2019 European Conference for Computing in Construction* 1, 341–50.

Bonduel, M. (2020). Including Widespread Geometry Schemas Into Linked Data Based BIM Applied to Built Heritage. *Proceedings of the Institution of Civil Engineers-Smart Infrastructure & Construction*, 1-16.

BrickSchema. (2016). “Brick Schema: A uniform metadata schema for buildings.” Accessed April 10, 2021. https://brickschema.org/.

Chen, S., Jin, R. and Alam. M. (2018). Investigation of Interoperability between Building Information Modelling (BIM) and Building Energy Simulation (BES). *International Review of Applied Sciences and Engineering* 9 (2), 137–44.

Corry, E., Pauwels, P. Hu, S., Keane, M, and O’Donnell, J. (2015). A Performance Assessment Ontology for the Environmental and Energy Management of Buildings. *Automation in Construction.*

European Telecommunication Standards Institute (ETSI). (2013). “Smart Applications REFerence Ontology, and extensions.” Accessed April 10, 2021. https://saref.etsi.org/.

Farzaneh, A., Monfet, D. and Forgues, D. (2019). Review of Using Building Information Modeling for Building Energy Modeling during the Design Process. *Journal of Building Engineering* 23, 127–35.

Gao, H., Koch, C. and Wu, Y. (2019). Building Information Modelling Based Building Energy Modelling: A Review. *Applied Energy* 238, 320–43.

Gilani, S., Quinn, C., and McArthur, J. J. (2020). A review of ontologies within the domain of smart and ongoing commissioning. *Building and Environment*,*182*(January),107099.

Gruber, T. (1993). A translation approach to portable specifications. *Knowledge Acquisition* 5 (2), 199-220.

Hinze, A., Heese, R., Luczak-Rösch, M. and Paschke, A. (2012). Semantic Enrichment by Non-Experts: Usability of Manual Annotation Tools. *Lecture Notes in Computer Science* 7649, 165–81.

International Organisation for Standardisation (2015). *Building Construction Organization of Information about Construction Works -- Part 2: Framework for Classification (ISO 12006-2).*

Kamel, E, Memari, A. M. (2019). Review of BIM’s Application in Energy Simulation: Tools, Issues, and Solutions. *Automation in Construction* 97, 164–80.

Mahdavi, A. and Wolosiuk, D. (2019). A Building Performance Indicator Ontology: Structure and Applications. *Proceedings of the 16th IBPSA Conference*, 77–82.

Negendahl, K. (2015). Building performance simulation in the early design stage: An introduction to integrated dynamic models. *Automation in Construction* 54, 39-53.

Pauwels, P., Corry, E., & O’Donnell, J. (2014). Representing SimModel in the Web Ontology Language. *Computing in Civil and Building Engineering*, 2271–12278.

Pauwels, P., Zhang, S., & Lee, Y. C. (2017). Semantic web technologies in AEC industry: A literature overview. *Automation in Construction*, 73, 145–165.

Pritoni, M., Paine, D., Fierro, G., Mosiman, C., Poplawski, M., Saha, A., … Granderson, J. (2021). *Metadata Schemas and Ontologies for Building Energy Applications : A Critical Review and Use Case Analysis*. 1–37.

Rasmussen, M., Lefrançois, M., Pauwels, P., Hviid, C. A. and Karlshøj, J. (2019). Managing Interrelated Project Information in AEC Knowledge Graphs. *Automation in Construction* 108, 102956.

Rasmussen, M., Pauwels, P., Lefrancois, M., and Schneider, G. (2021). “Building Topology Ontology.” Accessed April 10, 2021. https://w3c-lbd-cg.github.io/bot/.

Robert McNeel & Associates. *Rhinoceros 3D*. V. 7.4.21078.1001, 2021-03-19. Rober McNeel & Associates. Windows 10. 2020

Sadineni, S. B., Madala, S., & Boehm, R. F. (2011). Passive building energy savings: A review of building envelope components. *Renewable and Sustainable Energy Reviews*, 15(8), 3617–3631.

Sattler, L., Lamouri, S., Pellerin, R. and Maigne, T. (2019). Interoperability Aims in Building Information Modeling Exchanges: A Literature Review. *IFAC-PapersOnLine* 52, 271–76.

Stardog. (2005). “Stardog”. Accessed April 10, 2021. https://www.stardog.com/.

Stiny, G. (2006). III Using it to Design. In Stiny, G. *Shape*. MIT Press. Boston (USA).

Tchouanguem, D., Pauwels, P., Fonbeyin, H. A., Magniont, C., Karray, M. H., & Foguem, B. K. (2019). Integration of environmental data in BIM tool & linked building data. *CEUR Workshop Proceedings*, 2389, 78–91.

Terkaj, W., & Pauwels, P. (2017). A method to generate a modular ifcOWL ontology. *CEUR Workshop Proceedings*, 2050 (2017).

Web Ontology Language (OWL). n.d. “OWL Working Group: The Web Ontology Language.” Accessed April 10, 2021. https://www.w3.org/OWL/.

Zanni, M. A., Soetanto, R., & Ruikar, K. (2017). Towards a BIM-enabled sustainable building design process: roles, responsibilities and requirements. *Architectural Engineering and Design Management*, *13*, 101–129.

Classes and first degree Subclasses

OBO Classes	OBO Alignments	OBPD Classes	OBPD Alignments	OPA Classes	OPA Alignments
Building ●	BOT ifcOWL SM	**Document**		**Daylight**	
Collection		**Material ● ✚**	ifcOWL OBO SM	├ **DaylightExposure**	
├ **Envelope**		├ **EnergyPlusMaterial**		├ **DaylightSpatialAutonomy**	
└ **Interior**		└ **RadianceMaterial**		├ **GlareProbability**	
Element ●	BEO BOT ifcOWL SM	**MaterialAssembly**		└ **PointInTimeIlluminace**	
├ **AbstractElement ●**	BOT BRICK ifcOWL SM	└ Layer		**Energy**	
├ **AssemblyElement ●**	BEO ifcOWL SM	**Program ●**	SM	**EnergyAudit**	
└ **UnitElement ●**	BEO ifcOWL SM	├ **EnergyPlusBuildingProgram**		├ **BlowerDoorTest**	
Material ✚	ifcOWL OBPD	└ **EnergyPlusSpaceProgram**		├ **PFTAirInfiltration**	
Project ●	ifcOWL SM	**Schedule ●**	SM	├ **Survey**	
Site ●	BOT ifcOWL SM	**System ●**	BRICK SM	└ **ThermographicInspection**	
Space ●	BOT ifcOWL SM	**Zone ● ✚**	BOT ifcOWL OBO SM	**Evaluation ●**	SM
├ **Room**		└ **EnergyPlusZone**		**PerformanceObjective ●**	PF
└ **Zone ● ✚**	BOT ifcOWL OBPD SM			**SpatialComfort**	
				├ **PMV**	
				└ **AMV**	

● *External alignment* ***BEO:*** *Building Element Ontology* ***BOT:*** *Building Topology Ontology* ***ifcOWL:*** *IFC in RDF*
✚ *Internal alignment* ***PF:*** *Performance Framework* ***SM:*** *SimModelRDF* ***BRICK:*** *Brick Schema*

Figure 3: Table of OBO, OBPD and OPA Classes with Alignments to other AECO Ontologies.

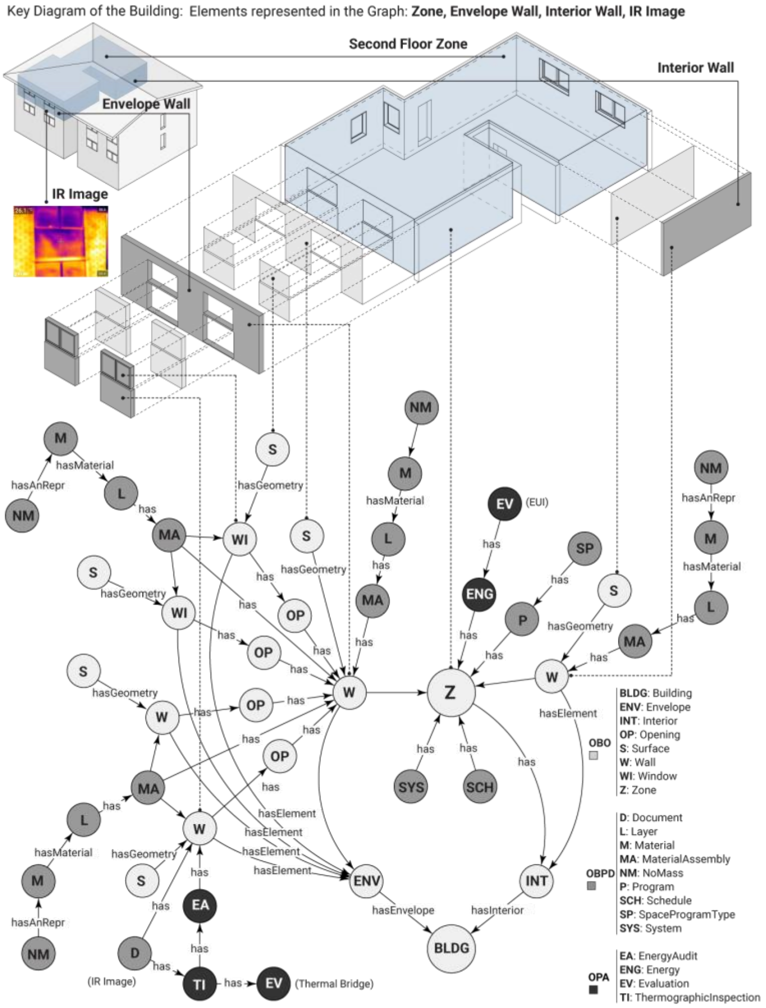
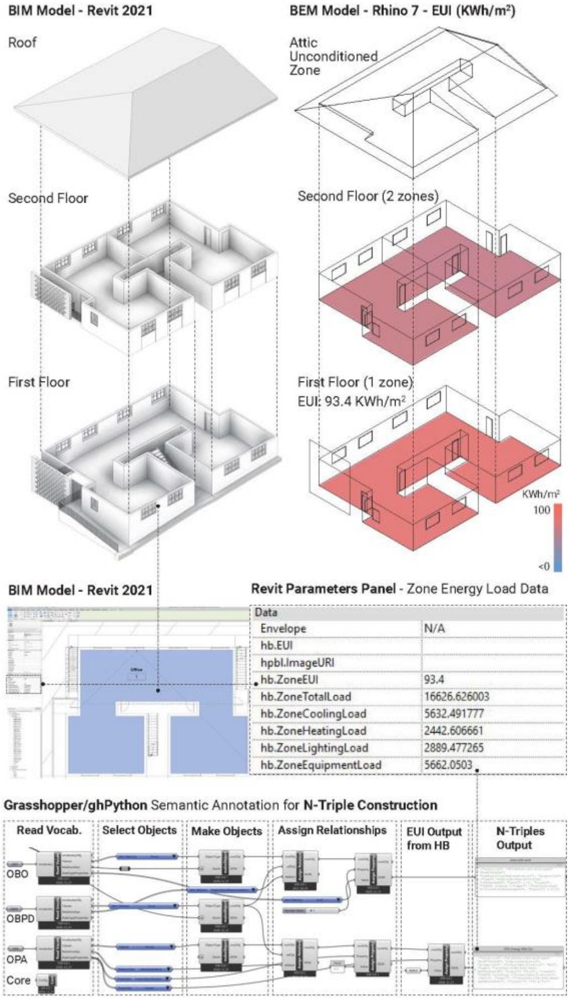


Figure 4: Conceptual graph representation describing the elements and relationships between a zone, an envelope and an internal wall, an IR image and zone energy load data, showcasing the three proposed ontologies, OBO, OBPD and OPA.



Data	
Envelope	N/A
hb.EUI	
hpbl.ImageURI	
hb.ZoneEUI	93.4
hb.ZoneTotalLoad	16626.626003
hb.ZoneCoolingLoad	5632.491777
hb.ZoneHeatingLoad	2442.606661
hb.ZoneLightingLoad	2889.477265
hb.ZoneEquipmentLoad	5662.0503

Figure 5: Top: BIM and BEM model representations of the case study model, Middle: Zone energy loading data generated in Ladybug tools are integrated in Revit's Parameter panel, Bottom: Semantic annotation system in ghPython showing the subject – predicate – object relationship assignment process and automatic conversion to N-Triple format for the zone EUI data.