Toward artificially intelligent cloud-based building information modelling for collaborative multidisciplinary design

Article *in* Advanced Engineering Informatics · August 2022

DOI: 10.1016/j.aei.2022.101711

5 authors, including:

Zijian Wang
Technion - Israel Institute of Technology

Boyuan Ouyang
Technion - Israel Institute of Technology

Duygu Utkucu
Istanbul Technical University

Siyu Chen
Imperial College London

TOWARD ARTIFICIALLY INTELLIGENT CLOUD-BASED BUILDING INFORMATION MODELLING FOR COLLABORATIVE MULTIDISCIPLINARY DESIGN

Revised and resubmitted: June 2022.

Rafael Sacks, Professor, Technion – Israel Institute of Technology[1]

Zijian Wang, PhD Candidate, Technion – Israel Institute of Technology

Boyuan Ouyang, MSc, Technion – Israel Institute of Technology

Duygu Utkucu, PhD Candidate, Technion – Israel Institute of Technology

Siyu Chen, PhD Candidate, Technion – Israel Institute of Technology; Researcher, Trimble Inc.

ABSTRACT

The technological tools people use for designing buildings have progressed from drawings to descriptive geometry, and from computer-aided drafting and design (CAD) to building information modelling (BIM). Yet despite their use of state-of-the-art BIM technology, the multidisciplinary teams that design modern buildings still face numerous challenges. Building models lack sufficient semantic content to properly express design intent, concurrent design is difficult due to the need for operators to maintain model consistency and integrity manually, managing design variations is cumbersome due to the packaging of information in files, and collaboration requires making-do with imperfect interoperability between application software. In response, we propose a 'Cloud BIM' (CBIM) approach to building modelling that seeks to automate maintenance of consistency across federated discipline-specific models by enriching models with semantic information that encapsulates design intent. The approach requires a new ontology to represent knowledge about the relationships between building model objects within and across disciplines. Discipline-specific building models are stored together with their data schema in knowledge graphs, and linked using objects and relationships from the CBIM ontology. The links are established using artificially intelligent semantic enrichment methods that recognize patterns of location, geometry, topology and more. Software methods that operate along CBIM relationship chains can detect inconsistencies that arise across disciplines and act to inform users, propose meaningful corrections, and apply them if approved. Future CBIM systems may provide designers with the functionality for collaborative multidisciplinary design by maintaining model consistency and managing versioning at the object level.

KEYWORDS: Building information modelling; Concurrent engineering; Design collaboration; Knowledge graphs; Semantic enrichment.

1. INTRODUCTION

Design, procurement and construction of modern buildings and other facilities require architects, engineers, managers, fabricators and builders with a wide range of professional specializations to collaborate closely. Practitioners use a variety of technologies to conceptualise, generate, communicate and store information. Over time, the tools available to practitioners in the Architecture, Engineering and Construction (AEC) sectors are improved through technological advances. The first major stepping stone in the development of graphical engineering communication as we know it today was the formalization of descriptive geometry by Gaspard Monge in 1794 (Migliari 2012). His work encapsulated the techniques of parallel projection and arrangement of orthogonal views of 3D objects on 2D media. Descriptive geometry became so deeply embedded in architectural and engineering practice, that the first applications of computer graphics to building design and construction took

[1]*Corresponding author: cvsacks@technion.ac.il*, Room 801, Rabin Building, Technion Campus, 32000, Israel

the form of computer-aided drafting (CAD), in which the mouse and screen were used to mimic working with pencil and paper to produce two-dimensional depictions of 3D solid reality. Although early computer graphics systems could represent both 3D and 2D geometry, the practice of documentation and communication using 2D drawings was (and perhaps still is) so deeply rooted in the construction industry, that sophisticated computer modelling tools were primarily used to generate drawings (Sacks et al. 2018).

The development of Building Information Modelling (BIM) represents a second major stepping stone, because it separated generation, compilation and storage of building information from its representation in documents. This was not a simple change of media, as was progress from paper to CAD, but rather a conceptual shift which allowed for compilation of an electronic prototype of a building. The prototype – the BIM model – encapsulates not only the geometric and alphanumeric information that defines the building, but also its function and its behaviour (Chandrasekaran and Josephson 2000; Lee et al. 2006). The ability to compile and test a prototype through virtual design and construction processes (VDC) has enabled much more efficient design and construction processes than were possible with CAD tools (Sacks et al. 2018).

Notwithstanding the advantages of current BIM processes, such as those standardised in ISO 19650 (ISO/DIS 19650 2018), they are constrained by the technology. Building models lack sufficient semantic content to properly express design intent; concurrent design is difficult due to the need for operators to maintain model consistency and integrity manually; managing design variations is cumbersome due to the packaging of information in files; and collaboration requires making-do with imperfect interoperability between application software. The current building design paradigm remains sequential: design and development procedures are implemented one after another (Yazdani 1999), and the handoffs from one professional discipline's work to another are done in large batches of information. The resulting iterative flows waste time and resources because substantial rework and significant efforts are required for coordination across domains (Oraee et al. 2019; Sierra-Aparicio et al. 2019).

Researchers and practitioners have envisioned much more tightly integrated design and construction practices (Fischer et al. 2017). Integrated Project Delivery (IPD), for example, has emerged as a contracting form that encourages concurrent design processes. The concept is that all the project team members work together and utilize the best collaborative tools (Ghassemi and Becerik-Gerber 2011). Many projects procured under IPD contracts adopt the 'Big Room' method that co-locates team members in a common office space, where they can work with one another to design in parallel. In the big room, people achieve better collaboration (Kent and Becerik-Gerber 2010), but information is still siloed as each discipline uses specialized tools for authoring and editing building models. Information exchange between disciplines in model files is imperfect and incomplete, requiring iterative rounds of clash checking and coordination.

In the manufacturing industry, Concurrent Engineering (CE) has developed as a method of managing engineering design and development in overlapping, concurrent processes (Safdari 2018). The process is streamlined through the effective integration of metadata into the design phase to maximize information exchange. CE appears to have many potential benefits, including shortened project life-cycle, enhanced product quality, and reduced manufacturing costs (Yassine and Braha 2003).

With the goal of supporting closely integrated virtual design with concurrent design, analysis, simulation and documentation, we propose a new paradigm, for the structure and functionality of integrated BIM platforms and tools. We call this new paradigm 'Cloud BIM' (CBIM). In CBIM, federated building models are stored in knowledge graphs rather than in disparate BIM files. Consistency across the set of distinct design domain models is maintained by functions that operate along chains of node-to-node relationships that express the interrelated behaviours of objects that represent design intent. These relationships are supplemented to the knowledge graphs through artificially intelligent semantic enrichment.

The remainder of this paper is structured as follows. Section 2 provides background on current limitations and barriers for design collaboration using BIM for AEC; potential benefits of closely integrated and concurrent engineering; design intent and constraint management capabilities of product lifecycle management software; and on the opportunities presented by BIM semantic enrichment techniques, linked data structures, graph representations and information storage and management in the cloud. Section 3 presents a new design communication paradigm through a set of practical use cases. Section 4 lays out the architecture of a CBIM system, section 5 details the CBIM ontology and section 6 presents experimental implementations that demonstrate basic feasibility. A discussion of technical considerations, advantages and limitations follows in section 7.

2. BACKGROUND

The introduction highlighted the need for designers from multiple disciplinary specializations to collaborate closely to design and construct complex modern buildings effectively and efficiently. The need for close integration is the first topic reviewed in this section. Against the backdrop of the potential of integration, we then outline the limitations of the workflows that have developed around the application of BIM for building design, critiquing those aspects in which the technology falls short in terms of supporting collaboration among diverse design disciplines.

Under the current state-of-practice, designers have advanced discipline-specific modelling, analysis and simulation tools, but the processes remain siloed and serial, with limited support for sharing information across disciplines. The state-of-the-art in cross-discipline interoperability is the set of export and import routines implemented by BIM software vendors in conformance with the IFC standards (IFC 2018). The resulting file-based exchanges fall far short of the intensive information exchanges that are needed for collaborative design. Recognising these limitations, researchers have developed various techniques that hold the potential to address them. These include alternative representations, such as linked data and the semantic web, semantic enrichment, and object-based cloud data storage to replace file-based exchanges. These are among the foundational technologies for software systems designed for the CBIM paradigm, and thus they are the subjects of the next sections in the background review.

2.1 Integrated, collaborative and concurrent engineering

Concurrent Engineering (CE), also known as Parallel or Simultaneous Engineering processes help designers respond to changes (Anumba and Evbuomwan 1997). As an integrated and concurrent approach to the design of products and their relevant processes, CE aims to achieve high-quality collaboration by considering all aspects of a product's life cycle from the beginning of design, including quality, cost, schedule and user requirements (Anumba et al. 2007). When implemented optimally, CE can reduce the need for and the extent of design iterations by providing detailed information from all disciplines on which to base design decisions, thus avoiding unnecessary rework (Ballard 2000). However, implementation of CE in construction is perceived to be more difficult than in manufacturing due to the deep organisational fragmentation in the design process (Akintoye et al. 2012). Researchers have highlighted the need for detailed exploration of existing best practices of design integration and collaboration to identify concepts or principles that may mitigate issues arising from fragmentation (Mohd Nawi et al. 2014).

The MacLeamy Curve (Guide 2007), shown in Figure 1, indicates a preferred design process in which designers shift effort to the early stages of design, when the cost of change is low. Design disciplines are engaged and integrated early, collaborating to develop more comprehensive yet well-coordinated design solutions. This runs counter to traditional siloed practice, in which each discipline prefers to wait as long as possible before engaging in detailed design, in the hope of minimizing rework as a result of changes made to the design by other disciplines. However, although BIM tools ease some of the design coordination problems, teams still struggle to fulfil MacLeamy’s vision. Data exchange and interoperability limitations are a barrier to closer collaboration in designing and delivering construction projects in the AEC industry (Afsari et al. 2017). People working on the same construction project using different software applications still complete fragmented tasks sequentially, although they would be better performed in parallel (Sacks et al. 2018).

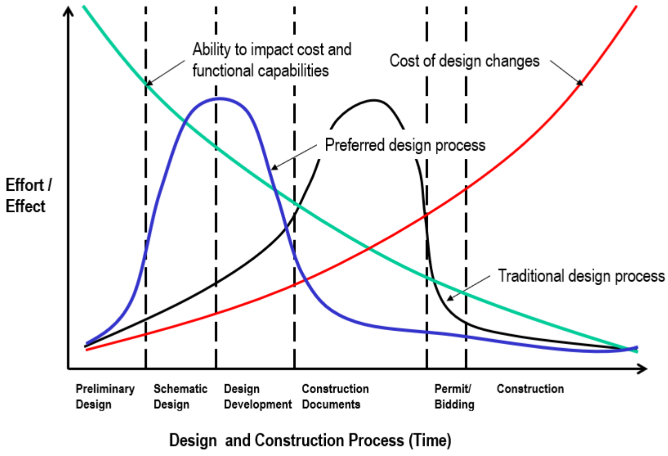


Figure 1. The MacLeamy Curve (Guide 2007).

2.2 Limitations to design collaboration and communication using BIM

Achieving close collaboration among professionals from different design and construction disciplines and organisations in building design is difficult (Matthews et al. 2018), even when BIM tools are used in construction projects (Oraee et al. 2019). Among the problems evident in the literature: high demand for meetings for coordination rather than creation of information (Sierra-Aparicio et al. 2019); interoperability problems among heterogeneous software tools (Lai and Deng 2018); compatibility issues among homogeneous software due to different versions; problems coordinating among design versions due to ongoing modification of the design (Sierra-Aparicio et al. 2019); and cumbersome tasks that result from manual changes made outside of the model information across domains (Isaac 2011). These problems can lead to misunderstandings, data misinterpretation, extensive rework (Zhao et al. 2017), cost overruns (Aljohani 2017) and delivery delays for construction projects.

The need to facilitate BIM collaboration on construction projects has become a central concern (Oraee et al. 2017). Multiple vendors now provide Common Data Environment (CDE) services in the cloud for sharing project information in accordance with ISO 19650. As file-based systems, they require each version of a design to be stored as a separate file. The more advanced systems allow users to work simultaneously on files through object-locking mechanisms and changes are logged. However, they have limitations on (i) tracking the relationships between recreated, merged, or split components to the original component and (ii) recognizing logical dependencies of design changes (Pilehchian et al. 2015). Maintaining subsets of building elements that are alternative versions of other subsets within the same file and logically performing the substitutions both within and across design domains is not supported. This has led researchers to propose graph based version control for BIM in support of asynchronous collaboration (Esser et al. 2021). Mattern and König (2018) for example, presented a conceptual approach to store object-level design options based on the IfcOption entity in the IFC 4.0 schema, and demonstrated retrieval from a Neo4j graph database.

A survey carried out by one of the major BIM software vendors among their clients found that practitioners would welcome a platform with a single repository and multi-user interfaces tailored for each design discipline (Franzke et al. 2021). In general, however, the wide distribution of legacy software products and the product-specific feedback collected from users motivate BIM software companies to pursue incremental product innovations (Acemoglu et al. 2020). Companies are unlikely to make radical changes to current systems because they may render their products obsolete (Escrig-Tena et al. 2021).

2.3 Design Intent and Constraint Management in Mechanical CAD Systems

In some commercial mechanical computer-aided design (CAD) systems, users can define geometric (e.g., collinearity, perpendicularity, symmetry, etc.) and dimensional (e.g., distance, angle, and radius) constraints between objects or object features. Parametric CAD systems evaluate the constraints using solvers that apply numerical methods, symbolic computing, graph-based analysis, or rule-based reasoning (Yoo et al. 2021; Zhou et al. 2020). The use of artificial intelligence in CAD/CAM systems has made design and manufacturing operations increasingly intelligent. This integration allows (i) for more complex designs, (ii) reduced need for manual trial and error, and (iii) more effective decision-making and design optimization.

Cloud-based tools and services have been implemented in product engineering design (CAD), analysis (CAE) and manufacturing (CAM) (Wu et al. 2015). PTC Inc.'s Onshape, for example, is a fully parametric cloud-based Software-as-a-Service (SaaS) product development platform (Onshape 2022). It incorporates CAD, data management, and analytics, allowing designers and engineers to collaborate concurrently. Design elements are stored as objects, thus removing the limitations associated with file-based version control. Dassault Systèmes' 3DEXPERIENCE cloud platform provides functions to develop and manufacture digital-smart products in a collaborative business process (Dalpadulo et al. 2020; Wu et al. 2017). These and similar mechanical CAD systems reflect design intent by establishing parametric and topological constraints in the product assemblies, product parts and their features. Although these tools are widely used in the manufacturing industry, their use in the AEC industry is rare and there is little effort invested in academia or in the software industry to develop tools with similar capabilities.

2.4 Linked Data for building design and construction

Buildings, their spaces, the elements and assemblies of which they are composed, and the construction processes applied to build them can be represented conveniently using linked data because much of the meaning of the concepts is formed by the relationships among them. The relationships are often essential in determining the form, function and behavior of the objects. Researchers have proposed multiple ways to use linked data in the AEC industry, including graphs in general and the semantic web in particular (Domingue et al. 2011; Pauwels et al. 2011). A major advantage is that the semantic web supports machine-understandable data descriptions and enables software to query and infer semantics from the data model. The Resource Description Framework (RDF), as a core technology of the semantic web, is a flexible and generic language to represent and combine information from diverse knowledge domains in a graph format that a computer can understand (Manola et al. 2004; Segaran et al. 2009).

The primary motivation for use of semantic web technologies in the AEC industry is to link multidisciplinary building models to ease interoperability across design domains, to support decision making by applying intelligent software agents, and to improve collaboration among participants (Abdul-Ghafour et al. 2011, 2014; Le and Jeong 2016; Törmä 2013; Yang and Zhang 2006). As BIM models are not stored in graph formats, the first step before linking models is to represent them as graphs. Le and Jeong (2016) designed a data wrapper for converting LandXML and CSV format files to RDF graphs, but these were intended for 2D information. Ismail et al. (2017) built IFCWebServer to transform IFC files to RDF graphs but excluded the geometry. The IFCtoRDF parser can compile IFC files to RDF graphs and covers all the information defined by the ifcOWL ontology (Pauwels 2021; Pauwels and Terkaj 2016). IFCtoRDF is simple to use, removing previously complex procedures, and has thus facilitated application of semantic web technologies in the AEC industry. However, the RDF graphs obtained from it are cumbersome, with deep chains of nodes and edges to express geometry and other attributes of building elements. The relationships between building elements in the STEP physical files (SPF) are mostly implemented as objectified inverse relationships, where an "IfcRel" type entity carries pointers to two or more related elements, making it difficult to traverse the resulting graph. These issues were partly addressed in the more recent IFCtoLBD converter (Bonduel et al. 2018), although this tool does not export any aggregation relationships.

A second pre-requisite to effectively linking multiple BIM models as graphs is an ontology defining the range of possible inter-domain relationships. Yang and Zhang (2006) suggested that building information schema should contain objects that represent relationships across disciplines and constraints. Records of operations on the models could also be included (Abdul-Ghafour et al. 2007). Törmä (2013) demonstrated how objects belonging to different design domains could be linked, but this was performed manually, which would be impractical and error-prone in any practical implementation. While these discussions outlined the possible contents of such an ontology, there has been no systematic research on the topic.

2.5 Semantic enrichment of BIM models

Semantic enrichment applies AI techniques to infer the existence of implicit objects, relationships, and attributes of BIM models and add them to the models, enhancing the quality of data and paving the way for further intelligent functions. In early semantic enrichment research, expert systems were used to classify bridge objects (Ma et al. 2018; Sacks et al. 2017). This demonstrated the feasibility of applying symbolic AI techniques to semantic enrichment but faced the limitations of complex rule sets with a narrow scope. Some tasks, like room type classification, deal with abstract concepts that a human easily understands but are difficult to process with expert systems. As a result, researchers have explored the application of supervised machine learning and deep learning algorithms to semantic enrichment. The algorithms recognize latent feature patterns from training examples, which helps capture implicit semantics that can hardly be expressed as simple, hard-coded if-then rules. Koo and his team attempted two different advanced techniques on a similar IFC object classification task, including machine learning algorithms (Support Vector Machine) (Koo and Shin 2018; Koo et al. 2019) and 3D deep learning networks (Koo et al. 2021a; b). The high accuracy achieved in the two experiments further proved the feasibility of adopting machine learning and deep learning algorithms for semantic enrichment. Recently, Wang et al. (2021b; a) applied graph neural networks (GNN) to predict room types of apartments based on BIM models expressed as graphs, thus broadening the palette of tools for semantic enrichment of BIM models.

Most semantic enrichment research has focused on classifying BIM entities within a BIM model belonging to a single design domain. Tasks that add explicit relationships among BIM entities, such as association and aggregation, have only been explored in limited applications and only with rule-inferencing (Bloch and Sacks 2020). These are particularly relevant in the context of linked data, where BIM models are represented and processed as graphs. Researchers have used common unique resource identifier (URI) to link objects across graphs for applications such as decision-making in highway asset management (Le and Jeong 2016), but this algorithm can only be applied to generate limited inter-domain relationships. Therefore, using semantic enrichment techniques to predict both intra-domain and inter-domain relationships automatically is an important future research direction.

3. IMAGINING A NEW PARADIGM

Given the complexities of modern buildings, with their multiple interdependent functional systems, close collaboration among integrated multidisciplinary teams is essential for their design and construction. As outlined in the literature review, the multidisciplinary federated BIM models that are currently the state-of-the-art technology solutions for this purpose have numerous drawbacks that necessitate extensive rework and long cycle times in compiling information for the buildings. The drawbacks include implicitly identified semantic contents and object relationships across disciplines that are inaccessible to software; inconvenient management of design variations in model files; poor interoperability across software applications. The key aim of this work is therefore to outline a system architecture and an ontology that support development of a new generation of innovative BIM tools that might ameliorate some of these problems.

The proposed system concept, CBIM, is designed to fulfil a new mode of collaborative design and detailing. The CBIM paradigm can be defined by the work processes it purports to enable. Working within a CBIM system, architects, engineers and construction detailers should be able to:

- Design and model: Use discipline specific software tools to pursue their individual design tasks, generating model geometry and product information with the building element concepts, relationships and behaviours they are familiar with. Designers work in parallel to generate a set of federated models.
- Maintain consistency: Receive an alert when a collaborator from a different discipline generates or changes information that creates design features that are incompatible with their own current representation of the building or violate a design requirement or constraint, whether spatial or related to design intent, for which they are responsible.
- Review proposed corrections: The system may propose action within their model or within the models of the other design domains to resolve any conflict. Users review, accept, modify, or reject the conflicting design change and or the action proposed by the system.
- Manage versions: Store multiple alternative versions of aggregations of objects within the design, which are fully coordinated with corresponding aggregations of objects in the models of other disciplines.
- Analyse: Review different combinations of design versions and run performance simulations and analyses to achieve global optimization of a design subject to multiple criteria.

Reflecting on the state-of-the-art in similar systems from mechanical engineering and on the broad experience of researchers and practitioners reported in the literature, we propose that achieving this functionality will entail object-based cloud storage; federated sets of discipline specific models; intelligent enrichment of models with semantic information that encapsulates design intent; enrichment of cross-discipline relationships that support automated real-time monitoring of model consistency; and intelligent control and maintenance of alternative design configurations at object-level. The following section (section 4) outlines a candidate system architecture devised to provide this functionality.

In this section, we propose two practical use case ‘vignettes’ to illustrate work within a CBIM system environment. The first use case concerns the propagation of design changes from one design discipline’s model to those of the other disciplines’ models. Inter-domain change maintenance enables individuals collaborating on a federated model to quickly receive information about design changes from other domains and to respond to the proposed changes in a way that maintains the logical integrity of the model and enables them to detail their designs concurrently.

The second use case concerns versioning. The idea here is to accommodate variants of the same objects, or aggregations of objects, within a model, making the collaboration more flexible and agile. These use case examples exemplify functionality that is not provided in currently available BIM software (Mattern and König 2018).

Both use cases are illustrated using a federated BIM model of a 17 story multi-purpose building constructed recently in Herzliya, Israel, depicted in Figure 2 (a). The use cases concern the stairwell of the south-east block, shown in Figure 2 (b). The BIM model had three component parts, each compiled and maintained by professionals from different disciplines – an architectural model, a structural model, and a Mechanical Electrical Plumbing (MEP) systems model.

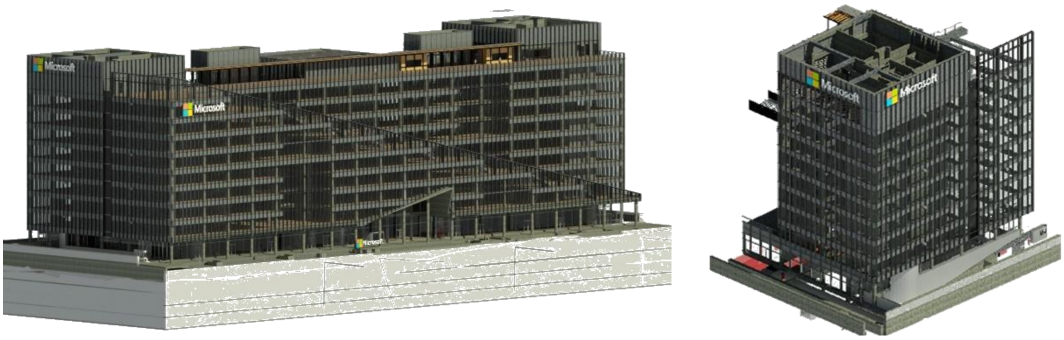


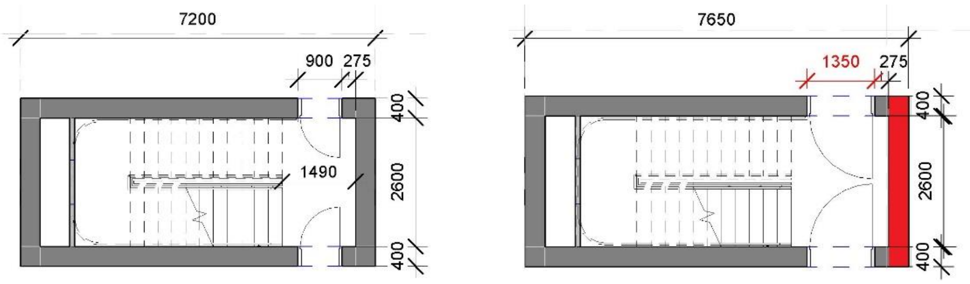
Figure 2. (a) the 3D federated BIM model, (b) the south-east block.

The use cases are based on actual design concerns within the building, but they are hypothetical in that they represent future work processes that were not possible during the project’s actual design process. We begin by assuming that at some time during the design detailing phase, the client requested that a zone of a particular floor should have the capacity to accommodate an auditorium. A design check for fire safety revealed that for this occupancy type, the width of the entrance doors to the stairwell (900mm wide) was insufficient to provide safe egress. The required width was 1,350mm. In the following, we trace the way in which a future CBIM system might support the designers from the three disciplines to collaborate in resolving the issues that arise.

3.1 Inter-domain change maintenance

The architects proposed making the doors wider by moving the eastern wall of the stairwell 450mm to the east, as illustrated in Figure 3. This was easily done in their BIM model. However, this design change affects the other discipline-specific models. In current best-practice for collaboration in detailed design, inter-domain change maintenance involves several functions: *assess*, *notify*, *propose*, *authorise* and *execute* (Rezgui et al. 1998; Stjepandić et al. 2015). The *assess* function determines if a change made in one domain requires adjustment in any other domain. For example, if an architect removes or changes the dimensions or opening directions of a window in a curtain wall with no impact on other systems, it is an intra-domain change. However, in our case, the eastern stairwell wall has a structural function and has sprinkler system pipes attached to it, and sprinkler heads located above the doors must be relocated.

Accepted for Advanced Engineering Informatics, June 2022 https://doi.org/10.1016/j.aei.2022.101711

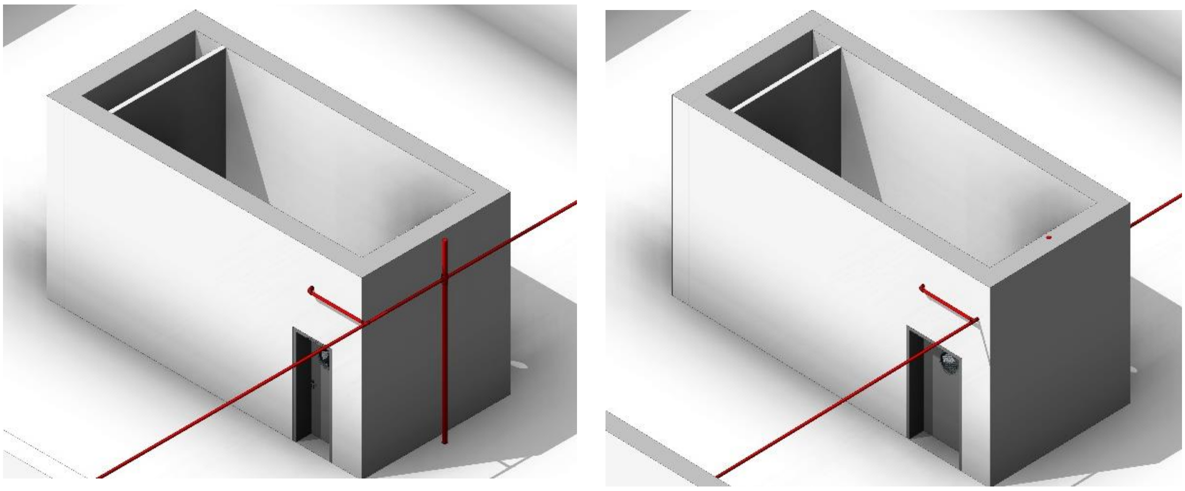


(a) (b)

Figure 3. (a) The initial design of the stairwell (b) Design change in the architectural model.

In current practice, assessment requires notification of all other disciplines, who must each review the change and determine any impact on their systems. This is commonly done in batches of design changes in asynchronous coordination. In a CBIM system, interdependence between system components is modelled explicitly using relationships that express the functional and topological interdependencies. For example, the architectural wall building element corresponds to the structural wall building element. When either one of the building elements in a correspondence relationship is altered, a software method of the platform can determine whether the alteration impacts the corresponding building element in the other system. Similarly, an ‘attached to’ relationship between the wall and the vertical sprinkler pipe triggers assessment of the impact of the relocation of the wall on the pipe location, determining that it must be repositioned too. A ‘centred’ topological relationship between the sprinkler heads and the width dimension of the door expresses the design intent and likewise triggers notification of a necessary change in the locations of the sprinkler heads. Since the change maintenance monitoring function of the platform runs in the background, it can notify the other designers of the impact of the architects’ change very soon after it is implemented.

The next step is for each affected discipline to propose an appropriate change in its own model to restore consistency to the federated model. For the sprinkler system, the MEP engineers need to relocate and reconnect the sprinkler heads and the associated pipes. The sprinkler and its vertical feeding pipes must be moved to new locations – the sprinkler 225mm to the east, as the design intent was for it to be centred above the door, and the vertical and horizontal feeder pipes 450mm to the east, as the design intent was for them to be adjacent to and fixed on the outer face of the wall. This is illustrated in Figure 4. Given the explicit inter-domain relationships, a software method can propose the changes needed, as it has the necessary geometric information about the changed architectural walls and the logic of the relationships. An MEP engineer may review the proposed changes in their model interface and submit them to the design manager for approval. If authorization is given, the changes are implemented in the MEP building elements. If not, or if consultation is needed, the system could initiate direct communication among the designers to coordinate the issue. Thus, the cycle of communication and model update can be essentially synchronous.



(a) (b)

Figure 4. (a) The initial design of the sprinkler system in the stairwell (b) Design intent inconsistency of the sprinkler system following architectural design change.

In the case of the structural design, the zone functional change required an increase in the design live load. The structural engineers propose to deepen the 200mm-thick slab by 50mm. In this case, the structural slab is the core layer within the architectural composite floor. A CBIM 'corresponds_to' relationship between the architectural floor BIM building element and the structural slab element triggers notification of the need for a change to the layered floor, as simulated in Figure 5. The architects may consider multiple alternative ways to adjust the design to accommodate the change. Notification includes the location of the change, a description, assessment of priority, and a list of the elements affected.

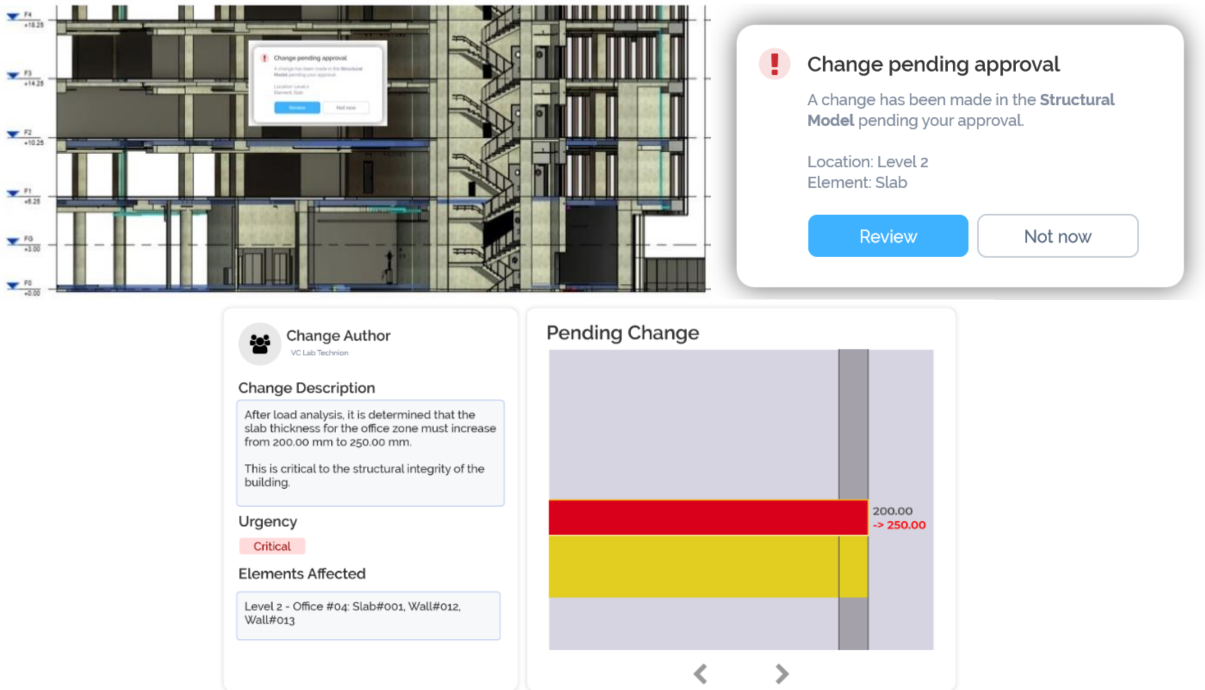


Figure 5. CBIM platform notification of a change to the thickness of a structural slab.

The "propose" function, an artificially intelligent design agent, provides options that account for design constraints that are also modelled explicitly in the CBIM platform model - for example, a design constraint defining a minimum value for the floor to ceiling clear height. Following this hypothetical scenario, the system may propose (a) raise the top face of the concrete slab by 50mm and reduce the thickness of the architectural flooring layer (originally 100mm) by 50mm to keep the floor level unchanged (see Figure 6); (b) lower the bottom face of the slab by 50 mm and reduce the depth of the false ceiling; or (c), adjust the top and bottom faces with dimensions that sum to 50mm.

Figure 6. A recommended action to implement the design change consistently across the structural model and the architectural model.

The "authorise" function allows designers to accept or reject a proposed modification. In the latter case, they would communicate directly and then each would apply the agreed changes to their own model. In this case, on its next consistency checking cycle, the CBIM platform would check that the resulting changes were indeed consistent, and whether they had further impact on any other systems. Note that in this example the solution to satisfy the changed design requirement – increased slab depth – could not be determined independently by either of the two design disciplines involved. Resolution required direct consultation because the constraints were interdependent. In cases such as these, synchronous coordination can resolve issues that would otherwise require multiple design iterations as each discipline made unilateral proposals for resolving the change. Furthermore, an intelligent agent which can read the constraints of both domains explicitly in the building model may be capable of identifying the interdependence, or even proposing a change that conforms to all constraints in a single step.

3.2 Object versioning

In situations such as the scenario described so far, a client or a designer may request an evaluation of the consequences of one or more possible design changes before committing to those changes. This is common, and in detailed design, it requires generation and maintenance of multiple design alternatives. In the example, the zoning change results in two versions for the floor thickness (300 mm and 350 mm, including structural slab and flooring layer), and two versions for the stairwell and door dimensions (door width of 900mm or 1,350mm with concomitant changes in the locations of the eastern wall and the sprinkler system elements). Consequently, there are four permutations of design options, as shown in Figure 7. In general, the number of possible permutations is an exponential function of the number of design changes and the number of possible conditions for each change.

As other researchers have highlighted (Nour and Beucke 2010; Zada et al. 2014), in the current use of file-based BIM models, maintaining any possible version requires saving multiple files. Researchers have extended the earlier work to propose systems for storing versions across federated discipline-specific BIM models (Mattern and König 2018; Pilehchian et al. 2015). A CBIM platform would adopt the same principles to enable local duplication and adaption of sets of building elements, labelling them according to the design versions to which they belong. Inter-domain relationships linking associated model objects that belong to the same version would thus allow filtering of building element objects from the federated model to reflect design permutations for review and analysis.

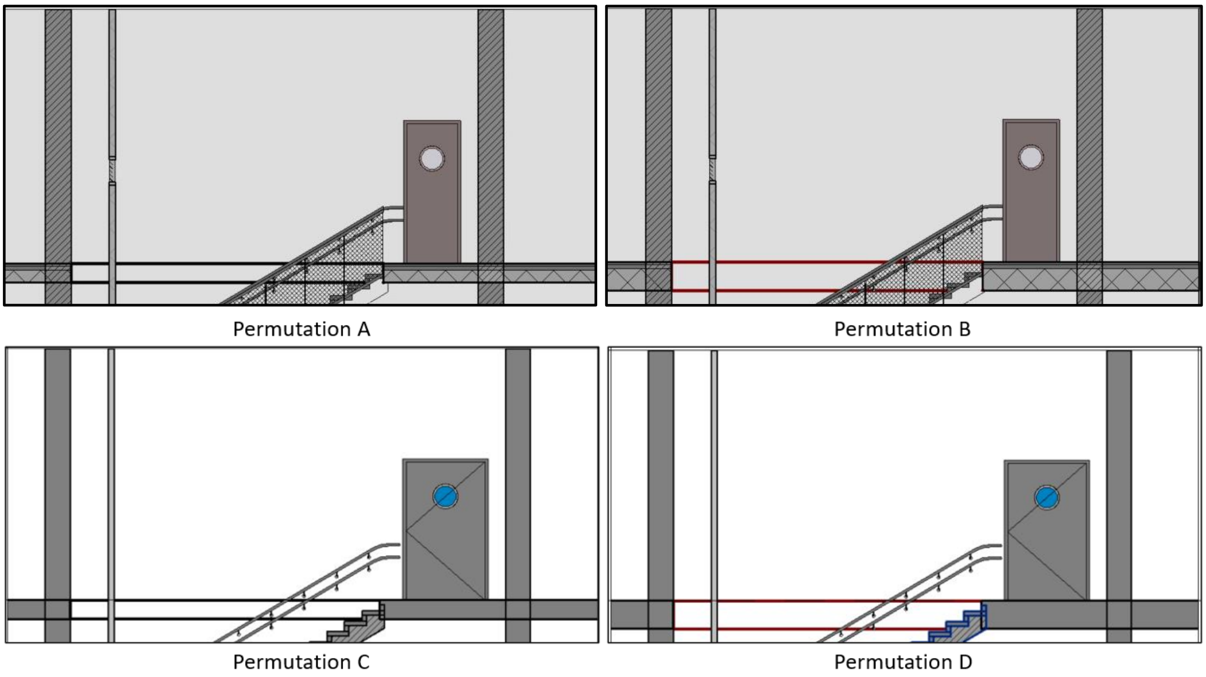


Figure 7. Four possible permutations of design changes for two slab thicknesses and two door widths.

4. CBIM SYSTEM ARCHITECTURE

Cloud-based BIM is centred around a graph database which stores multidisciplinary BIM models as sub-graphs in the cloud. Each sub-graph conforms to a discipline-specific ontology. A CBIM 'meta graph', with nodes and edges that link nodes across and within sub-graphs, explicitly expresses the relationships that are understood to exist between nodes in the different models that represent building elements and assemblies of elements that form functional building systems. The relationships include spatial and topological relations, correspondence of elements, design constraints and expressions of design intent. Making this web of relationships explicit may enable the desired intelligent functions that ease the interoperability problem, improve collaboration efficiency, ensure compatibility and consistency across discipline models, and ultimately improve construction quality, duration and cost. Figure 10 illustrates an example of a CBIM meta graph.

Figure 8 depicts the CBIM system architecture. The architecture adopts the convention of separating the system into three tiers for flexible and reusable implementation: a data storage tier, a tier of logic functions, and an interface tier. The components shown as blue rectangular symbols at the top and the centre of the figure are the CBIM system itself. These modules instantiate and maintain the meta graph – they are largely transparent to the architects and engineers who use the system, operating primarily in the background. The components labelled ① and ④ are discipline-specific software modules that integrate with the CBIM system. The cloud database in the centre of the figure comprises the discipline-specific sub-graphs and the CBIM meta graph.

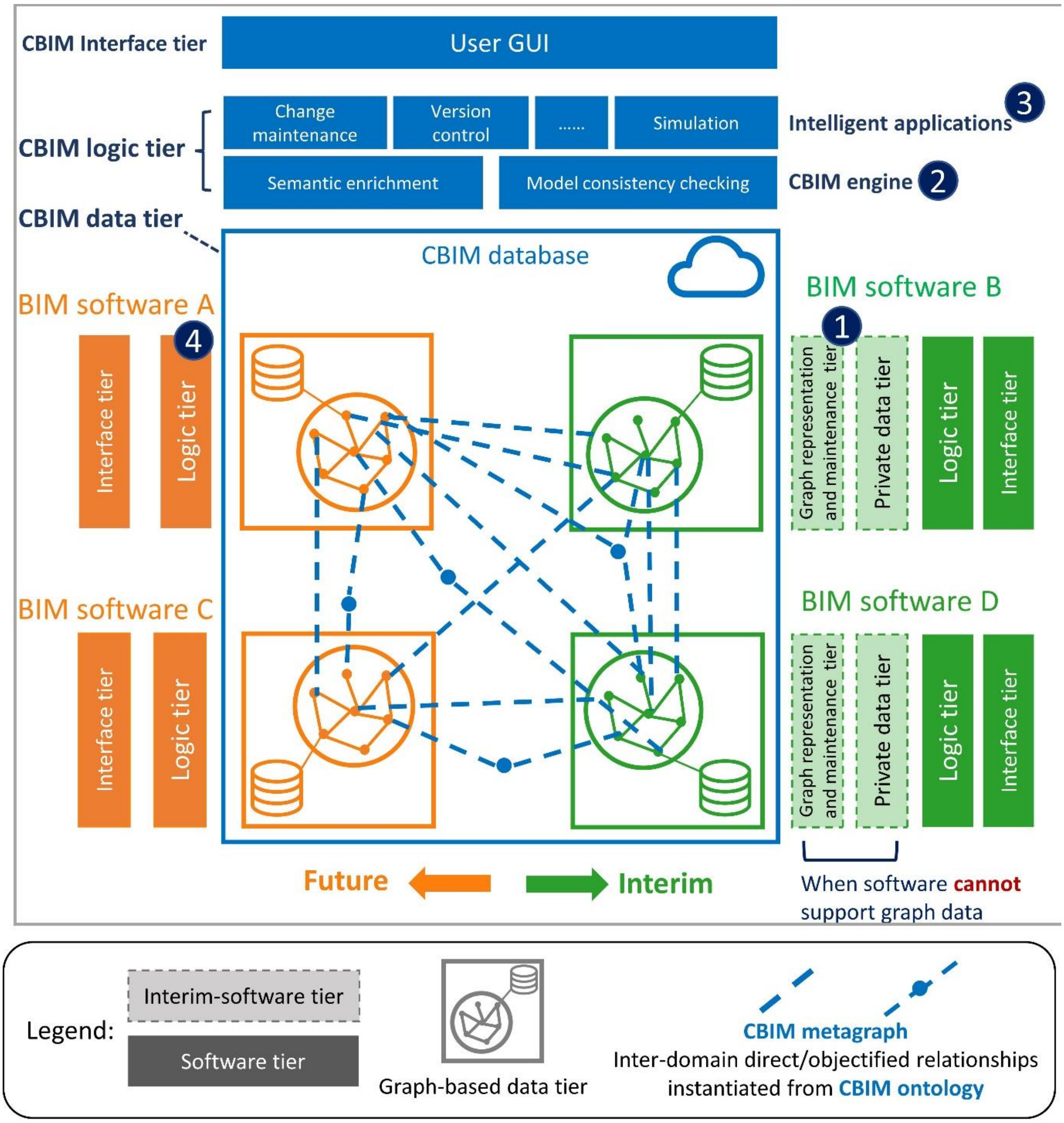
The applications labelled ① in Figure 8 represent an interim state in which legacy BIM systems are supplemented with an interfacing API software that mediates between the domain sub-graph in the CBIM cloud and a local application-specific data store (shown as symbols with dashed-line borders). This requires mapping between the open domain-specific data schema and the internal native schema of the application. Future BIM systems (labelled ④) may provide this functionality internally, thus obviating the need for an external translation function using the API.

When a project team starts a BIM project, the different professional disciplines generate domain-specific BIM models. These are compiled as sub-graphs in the CBIM database in the cloud (represented by the orange and green graph blocks within the CBIM database symbol, at the centre of Figure 8). A dedicated 'CBIM Engine' (labelled ② in the figure) runs continuously in the background with two main functions: semantic enrichment and model consistency checking. Consistency checking identifies conflicts or inconsistencies across models. Semantic enrichment supplements implicit building objects, relationships, and attributes to models. Semantic enrichment instantiates and maintains nodes and edges that represent semantic relationships between building elements across the domain-specific sub-graphs in three aspects: spatial relationships between building element objects; correspondence and other logical relationships among building elements; and constraints that can be inferred to make design intent explicit. The nodes and edges in the CBIM meta graph, shown in blue in the central part of the figure, are defined in a CBIM ontology. The engine and ontology are detailed in Sections 5 and 6 below.

The upper section of the logic tier contains intelligent functions that run directly on the linked graphs stored in the database. These include core CBIM functions such as change maintenance among multiple disciplines and version control, and peripheral applications such as generative design, analyses and simulations, etc. These software modules exploit the explicit representations of the relationships among building elements that are encapsulated in the meta graph to implement software that can perform operations on the federated models that are currently performed by human designers applying their implicit expert understanding of buildings and their behaviors. For example, if two data objects (a node or a graph section), one in each of two discipline sub-graphs, representing the same building element (such as an architectural and a structural representation of the same concrete wall) are geometrically inconsistent, then an intelligent conflict resolution software could prompt the two designers to resolve the conflict, and even propose a default solution. Communication with the users is routed through their domain-specific interfaces. Similarly, an integrative analysis function such as construction cost estimation could draw on information from multiple sub-graphs and the unifying information in the meta graph to produce a cost estimate which accounted for all building elements without erroneously including elements that may be represented in more than one sub-graph, thanks to the explicit correspondence relationships between duplicated elements. Automatic version control is also achieved on the meta graph. The version control function has access to read and modify the CBIM database. Whenever there are modified data, it keeps the original objects and relationships and supplements new building element nodes and graph sections with new properties that signify specific versions. Thus, a version history is maintained and designs can be traced backwards.

A core benefit of the CBIM concept lies in explicit semantic linking across sub-graphs. When inter-domain relationships are generated to connect multidisciplinary BIM models, the added layer of information enables development of software modules that can perform functions that require a more meaningful representation of buildings than is possible in current state-of-the-art BIM software. The linked data are the basis for the development of further desired intelligent functions. Graphs are a natural format for representing links between objects, and they also enable the application of advanced AI techniques, such as graph neural networks (GNN), to implement aspects of the CBIM engine and for building analysis and simulation software.

BIM data are complex, and graphs are not the most efficient medium for all BIM data. Although graphs can efficiently represent building object data and the latent relationships between BIM objects, they can only redundantly represent the complicated exact geometry of a model. Geometry may be represented as constructive Solid Geometry (CSG) objects and relationships or as Boundary Representation (Brep) meshes, but both require very large numbers of nodes and edges. BIM model graphs in which a large majority of the nodes were devoted to representing geometry would be unsuitable for learning using AI tools such as GNN. An effective alternative is to keep explicit geometry in modular formats or in sub-graphs external to the central BIM model graph and linked from building model objects within the graph, but at the same time to substitute the exact geometry with proxy nodes and edges that represent their physical manifestations and the logical relationships between them. For example, one may maintain a minimal bounding box for each object and supplement it with explicit spatial topology relationships that are derived from the exact geometry and that can be used for reasoning on the graph. Function calls from within the CBIM engine can use tried and tested geometry manipulation routines in legacy geometry manipulation libraries, such as ACIS (Spatial 2021), to compute the values of the relationships and supplement the CBIM graph with them.



455 *Figure 8. The system architecture of a CBIM environment.*

5. CBIM ONTOLOGY

Linking BIM models from different design domains requires a set of concepts that clearly define the semantics of the relationships that can be considered to exist between the objects in the models. For example, the building element objects representing a column in an architectural model will be of a different type, with different properties and property values than the objects representing the same column in a structural engineering model. Professionals understand implicitly that the disparate BIM objects represent the same physical column, but for software to make use of this attribute, it must be made explicit. Similarly, there are relationships between building elements that express design intents that are obvious to design professionals, but entirely absent from standard BIM model representations. For example, a water pipe designed to pass through a concrete wall embodies a well-understood design intent that there should be a hole penetrating the wall to accommodate the pipe. Furthermore, there are spatial and topological relationships between building elements that professionals infer on sight, but are inaccessible to software unless made explicit with appropriate nodes and edges in a model. The CBIM ontology described in this section is an attempt to define a minimal set of concepts to represent these inter-domain and intra-domain relationships that are essential to enable the intelligent functionalities of the CBIM paradigm.

An ontology can be defined as a "*formal, explicit specification of a shared conceptualization"* (Studer et al. 1998). Ontologies allow conceptual entities, properties, and the relationships among them to be systematically established through formally described logical statements. They are an accepted way to express knowledge in a machine-understandable manner while also being easily extendable and sharable. Numerous ontologies have been devised for the AEC industry, including building geometry (Wagner et al. 2022), topology (BOT - Rasmussen et al. 2021), smart applications and automation devices (DogOnt - Bonino and Corno 2008; SAREF - Daniele et al. 2022; ThinkHome - Reinisch et al. 2010), building systems (Brick - Balaji et al. 2016), and standard IFC data schema in web ontology language (IfcOWL - Pauwels and Terkaj 2016). Therefore, our starting point in defining the CBIM ontology was to adopt existing domain-specific ontologies as a basis. In particular, we reuse classes and relationships defined in the IFC schema and included in IfcOWL, such as those for building objects (*IfcBuildingElements*) and their child classes. As such, the CBIM ontology was designed as an expert-knowledge based, systematic and formal abstraction of possible relationships between objects within and across discipline-specific ontologies, that aims to express their intended interactive behaviours.

The CBIM ontology considers two kinds of explicit relationships between building objects: *spatial relationships* and *design intent relationships*. Spatial relationships refer to the objective descriptions of the topological relationships between building objects that an expert can infer from observing the geometries of the objects in the discipline models. For instance, "pipe P3 in the MEP model is adjacent to and parallel to beam B4 in the structural model", or "that column C1 in the structural model has an outer face aligned with the outer face of wall W2 in the architectural model". Although knowledge of this kind of factual object relationship is essential for numerous possible intelligent applications, no existing BIM solutions nor data schema are capable of representing and storing them explicitly. We propose that they should hence be explicitly stored in a structured way in CBIM systems, rather than being computed repeatedly whenever required. Details of the implementation of routines to compute their values are provided in Section 6.3 below.

Design intent relationships, on the other hand, describe latent design rules using parametric dimensional and topological constraints that bind the behaviour of building objects. These relationships express a desired or required state, to which objects in a model may or may not conform. As such, they are distinct from the spatial relationships, which arise from the current state of a model's objects. Design intent relationships may be instantiated through semantic enrichment software modules based on expert knowledge of the general design principles. Currently, three sources of knowledge are identified as the potential sources for the generation of design intent relationships: 1) building codes, from which clauses that formally describe design requirements can be translated into explicit rules, 2) AI learning, by which implicit patterns of design conventions could be recognized and formulated through examining multiple models of similar types, and 3) users' initiative, such as those from aesthetic or functional aspects.

The CBIM relationships can be represented in one of two ways: as direct links between object instances, or objectified as distinct instances with their own properties and linked to nodes representing the related instances. Using both is possible and results in a hybrid approach. The direct approach is appropriate where relationships are semantically straightforward without the need for attributes, while the objectified approach is a better fit for more complex relationships that need to be expressed with associated attributes and properties. *CBIM:CorrespondsTo* is a good example of the former, where the relationship only needs to describe the fact that different building object instances from two or more discipline-specific models are simply different abstractions of the same real-world physical manifestation. *CBIM:RelSpatial* is a perfect example for the latter, where the different topological position properties should use values enumerated from a property class attached to the relationship class.

Figure 9 depicts the current version of the CBIM ontology and a formal definition is provided as an OWL file in the dataset (Section 10). This formulation was devised from first principles and the knowledge accumulated in the literature, and refined through experimental development of software modules for inter-domain change management (section 3.1). The goal is to encapsulate expert knowledge concerning building design in a stand-alone ontology, and to demonstrate how this ontology can be applied. There are four basic object types: *elements*, *relationships*, *attributes* and *enumerated value sets*. Elements are building objects, and they adopt the definitions of *IfcElement* and the classes that inherit from it. Relationships have two types, *CBIM:RelSpatial* and *CBIM:RelDesignIntent*. A *CBIM:RelSpatial* relationship stores information about the topological relationship between any two objects. It has three attributes: *Topology*, which is an enumerated value as defined by Egenhofer (1989); *Offset*, a distance dimension; and *InContact*, which is a Boolean value. *CBIM:RelDesignIntent* instances are defined as *CBIM:RelConstraints* and have five properties – the physical connection type (if any), a description, the constraint's direction, its operator and its value.

For example, the vertical sprinkler pipe in Figure 4(a) was placed adjacent to the wall of the stairwell. The intention of the MEP engineer was clearly to place the pipe at a small distance from the face of the wall – large enough to allow for pipe inspection and maintenance, yet minimal to avoid obstructing the functional space. Using the CBIM ontology, the design intent can be represented as a set of parametric constraints between the pipe and the wall objects. The pipe and the wall belong to separate domain sub-graphs. Specifically, two *CBIM:RelConstraints* instances are instantiated between the objects, one for the minimum distance and the other for the maximum, with the edges *CBIM:hasSubject* and *CBIM:hasObject* pointing toward the pipe and the wall respectively. The connection type property is <BRACKET> for both constraint instances, the descriptions are the respective verbal explanations of the dual intents, the directions are the normal vectors to the outer face plane for the wall (global x-axis direction in this case), the operators are <LARGER_THAN> and <SMALLER_THAN> respectively, and the values are 20mm for the minimum and 50mm for the maximum. A detailed illustration of the instantiated constraint relationships is shown in Section 6.3 below.

6. CBIM SOFTWARE COMPONENTS

In this section, we describe the primary components of the software required to maintain and operationalize a CBIM environment. These include tools for generating graph representations of sets of BIM models (labelled ① in Figure 8), for instantiating and maintaining CBIM relationships between objects of the various initially disjoint sub-graphs generated from the domain-specific models that form the overall federated building model (the 'CBIM Engine', labelled ② in Figure 8), and the suite of intelligent tools that can operate on the full graph to maintain logical integrity, control alternative design versions and apply intelligent design aids and functional simulations (labelled ③).

For the purpose of explanation, the scope of the examples is limited to the three discipline-specific BIM models that served for explanation of the use cases in Section 3 above: architectural, structural, and MEP models. In full scale implementation, the scope would include models of all design disciplines.

Accepted for Advanced Engineering Informatics, June 2022 https://doi.org/10.1016/j.aei.2022.101711

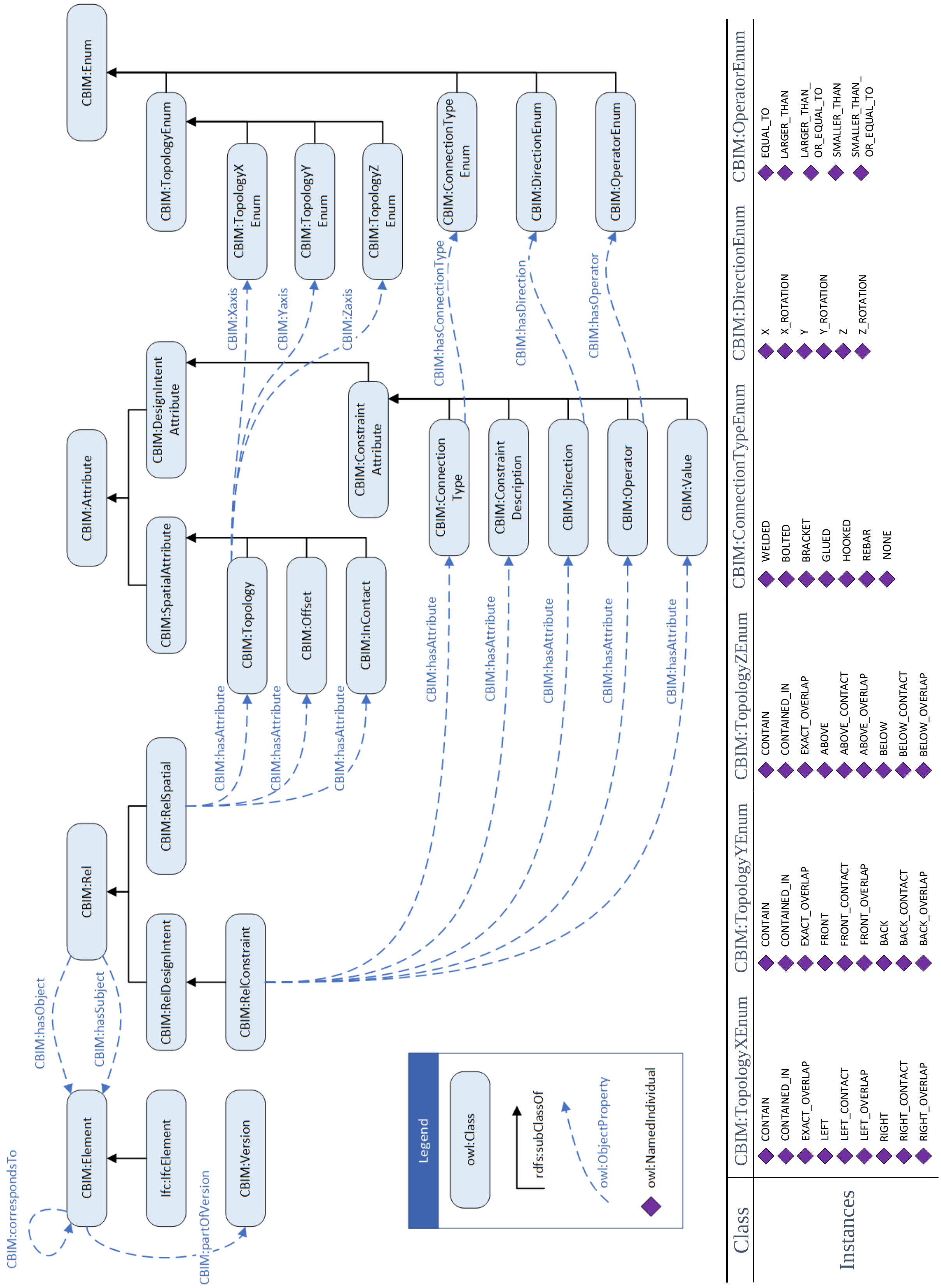


Figure 9. Principal concepts in the CBIM Ontology.

6.1 Graph generation and maintenance of domain models

The CBIM system architecture specifies representation of building models using graphs stored in the cloud. In a future state, discipline-specific design software applications may read and write information directly to and from their sub-graph representations (functions labelled ④ in Figure 8). However, current BIM software applications operate with files that store data in proprietary formats according to their native schema. For purposes of experimentation and validation it is therefore necessary to provide tools to generate graph representations of models compiled in stand-alone applications and to append them as sub-graphs in the cloud. Subsequently, additional functions are needed to propagate changes made in either the native model environment or in the CBIM environment to the corresponding representation. This section outlines how these tools might function (applications labelled ① in).

There are at least two different possible methods for generating graph-based building data. A simple pipeline method applied in the course of this research is to export the building model to IFC format, convert it from IFC to RDF using the IFCtoRDF parser (Pauwels 2021), and then upload the RDF file to a cloud graph database server. This is effective, but impractical for any use other than experimentation because it is limited to one-way exchanges and to use of the IFC schema.

A better approach is to write an application within the API of the native BIM software that can generate and upload the nodes and edges directly into the graph database. With this approach, the native and graph representations remain linked but Object IDs and the API software can update the native models in response to changes. In this work Dynamo scripts were prepared to export architectural or structural models from Autodesk Revit, and RDFlib, a Python library, was adopted to compile models as RDF graphs stored in CBIM database. Using existing functions in Dynamo that can change attributes of objects in Revit models, we implemented modules for direct communication between Revit and the CBIM graph, so that changes made in either can be propagated from one to the other. For lack of more specific domain ontologies, in this implementation the IFC schema was used for the sub-graphs.

6.2 Data format and storage considerations

There are two main categories of graph database tools, those that adhere to the rules of RDF and others that use more flexible constructs. GraphDB and RDFLib are examples of the first category. GraphDB has a user friendly interface for displaying, manipulating, and retrieving information from RDF graphs through SPARQL commands (GraphDB 2021). RDFLib, as a pure Python package, provides more flexible ways for users to construct and query RDF graphs by direct coding in a Python environment (Krech 2013). Using RDF as the graph storage format has two obvious advantages: RDF graphs enable the use of semantic web techniques such as SPARQL querying and linking of entities, and they are considered to be convenient and flexible for data exchange.

However, these tools, and RDF graphs in general, have limitations when enriching the semantics of graphs. AI techniques for semantic enrichment leverage latent patterns by computing feature vectors. Hence, one requirement for applying AI methods to enrich the graph models is to have nodes and edges with the ability to store feature vectors and matrices for computation. Due to the restriction of RDF construction rules, feature vectors and matrices cannot be attached to nodes in RDF graphs conveniently. Moreover, graph tools like GraphDB and RDFlib are designed to store graph data but lack the features necessary for graph computation. Property graphs are more suited to such computation. A possible solution is to separate the data storage in RDF format from the computation by converting data from the RDF graphs to property graphs in a pre-processing step, run the machine learning algorithms for semantic enrichment on property graphs, and supplement the results back into the RDF data repository. The property graphs converted from RDF are temporary, and not part of the persistent storage.

6.3 Linking disjoint sub-graphs

Data links between objects belonging to different domain-specific sub-graphs are the mechanism by which the CBIM paradigm confers the ability to propagate design changes and perform other intelligent functions on a ‘virtual’ federated multidisciplinary building model. The data links are specified in the CBIM ontology, and they express the intended behavioural or design logic of the building. They consist not only of edges, but also nodes for objectified relationships. The links are instantiated or removed whenever building element objects are appended, modified or deleted in any of the sub-graphs.

Linking begins with execution of basic volumetric functions to establish the CBIM spatial relationships between pairs of building element nodes from different sub-graphs that are adjacent to one another and or in contact or overlapping with one another. In the first instance, object geometries are compared using their bounding boxes to establish whether they are directly adjacent and if they may be intersecting. If they may intersect, their detailed geometry is retrieved and a volumetric Boolean clash check is run. Thus, for example, a *CBIM:RelSpatial* relationship would be established between the wall and pipe from the stairwell fire sprinkler pipe example, with attributes for *CBIM:Topology* = RIGHT, *CBIM:Offset* = 25mm, and *CBIM:InContact* = FALSE, as shown in Figure 10.

In the next step, AI procedures for semantic enrichment identify and instantiate CBIM nodes and edges that express design intent. They may use rule sets (as demonstrated in the examples in section 6.3 below and in the data set provided in Section 10) or graph machine learning algorithms for classifying relatioships. These methods transform the set of sub-graph representations of discipline-specific BIM models into an enriched, consistent, and interlinked meta graph upon which intelligent applications operate. Consistency checking may eliminate errors induced by human negligence or miscommunications during the design that lead to mismatched models. Semantic enrichment also supplements missing and inferred information for intra-domain sub-graphs for later use.

Both steps are triggered whenever data is added, edited or deleted from the CBIM database. They form the core of the CBIM logic tier shown in Figure 8.

6.3.1 Intelligent clash checking

Clash checking can be implemented in two stages. Any pair of objects that have a hard clash (those with *CBIM:RelSpatial* edges with topology attributes that indicate intersection) are checked for the following conditions:

1. The design intent is for one of the objects to penetrate the other object through an opening in the latter object, such as a pipe passing through a concrete wall. Appropriate rules can identify this situation based on the functions of the objects and can flag for user intervention to insert a void in the penetrated object, such as a pipe sleeve. An intelligent function could propose a solution to the designers for approval.
2. The design intent is that the two objects are alternate representations of the same physical object, such as a concrete wall object in a structural model that forms a core layer of a multi-layered composite wall object in an architectural wall. Here too, appropriate rules can identify the situation, check that the geometries and locations are consistent with this interpretation, and instantiate a *CBIM:CorrespondsTo* relationship (also shown in Figure 10).
3. The logic of the situation cannot be determined. This may occur where the various discipline designs are incompatible, such as a situation in which an MEP duct intersects with a structural beam. These cases can be brought to the attention of the building's designers for resolution.

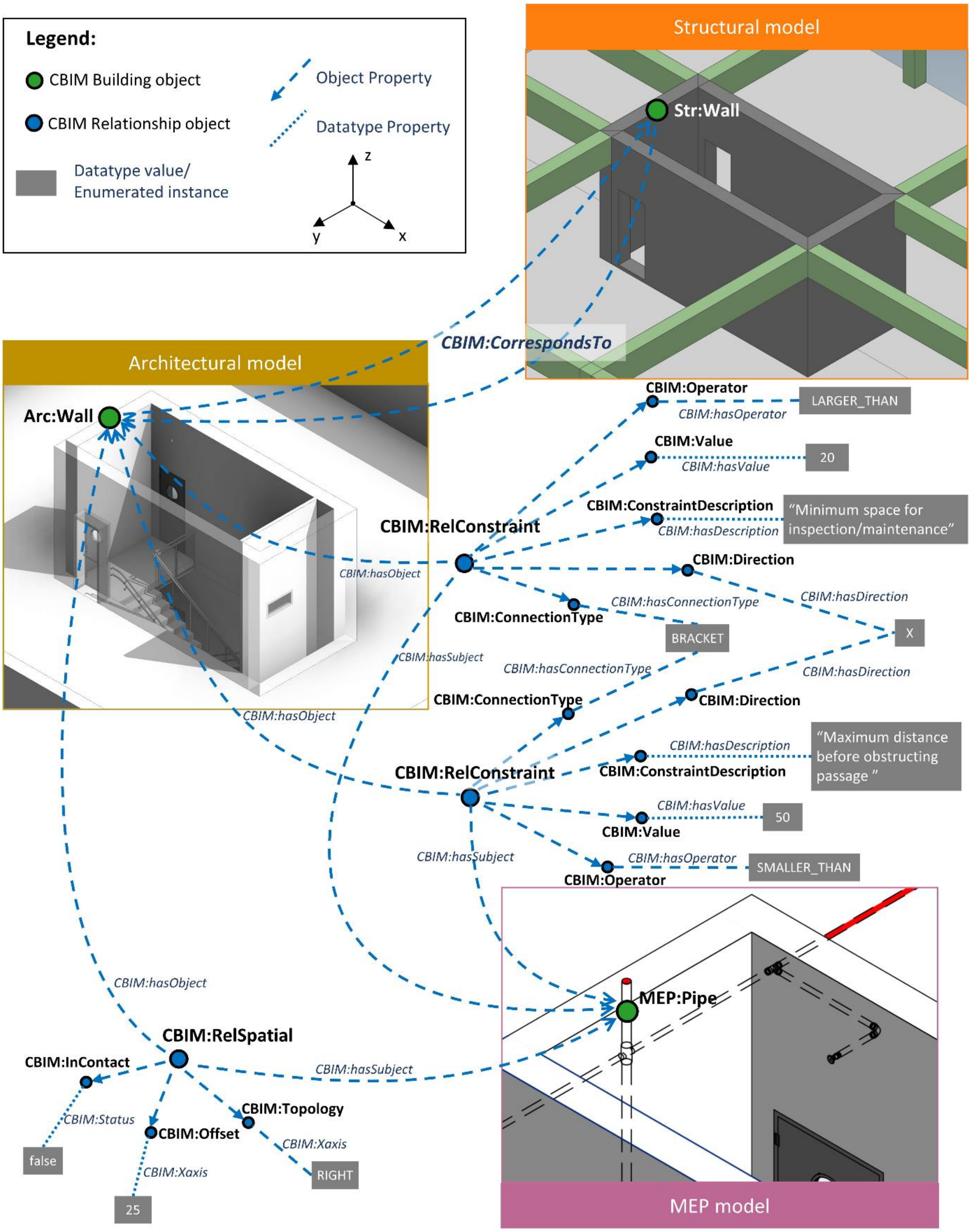
Note that a similar procedure could be performed for situations where there are soft clashes, i.e., where objects do not physically intersect but their proximity is such that it violates rules concerning constructability, maintainability, or others. These situations can also be identified rapidly based on combinations of features represented by the attributes of the *CBIM:RelSpatial* relationships. In a future situation where inconsistencies were flagged and labelled by users in multiple models over time, machine learning approaches could be applied to build new functions to identify, classify and possibly remedy a variety of such problems.

6.3.2 Semantic enrichment

Here, the system attempts to identify situations in which the design intent is implicit and to instantiate constraints that represent the design intent explicitly. Routines to derive this information may use symbolic AI in the form of rule inferencing, supervised machine learning, or deep learning, where, over time, accumulated archives of building model graphs can provide large data sets for learning. The design intent relationship between the vertical sprinkler pipe and the concrete wall of the stairwell, described at the end of Section 5, exemplifies this intent. Another example is that the doors generate openings in the architectural walls, and they require corresponding openings in the structural walls. There are four relationships here: two discipline-specific relationships handled by the local BIM interfaces (door with opening in architectural wall, opening in structural wall) and two handled by the CBIM engine (corresponding architectural and structural walls, and corresponding architectural and structural openings).

Figure 10 illustrates a CBIM meta graph consisting of three domain-specific graphs and the CBIM relationships that pertain to the spatial and the design intent constraint relationships between the eastern structural wall (*Str:Wall*), the eastern architectural wall (*Arc:Wall*) and the vertical sprinkler pipe (*MEP:Pipe*). In accordance with the structure and constraints defined in the ontology, inter-domain relationships instantiated by the CBIM engine may link building element nodes directly, such as *CBIM:CorrespondsTo* which relates *Arc:Wall* and *Str:Wall* in Figure 10, or as objectified relationships with intervening nodes, such as the *CBIM:RelSpatial* relationship between *Arc:Wall* and *MEP:Pipe*. The two *CBIM:RelConstraint* relationships determine that the pipe is fixed to the wall with brackets at a distance d such that $20 > d > 50$. Note that the brackets are not modelled explicitly at this level of detail – if they were, they would feature as building element nodes in their own right, with their own relationships to the wall and the pipe. The pseudo-code shown in Figure 11 illustrates the approach implemented to demonstrate semantic enrichment to establish inter-domain direct links of type *CBIM:CorrespondsTo*. The process has two steps: 1) identify and instantiate relevant *CBIM:RelSpatial* relationships, and 2) for all related elements in contact or overlapping with one another, determine possible correspondence and instantiate the direct links. In the first step, a module of the CBIM engine traverses all possible pairs of building element nodes from different sub-graphs. The bounding box geometries are compared. If they overlap one another in any of the three axis directions, a *CBIM:RelSpatial* relationship node is instantiated and linked to the subject and object building element nodes. The module then applies a Boolean solid intersection test using exact geometry retrieved from the building elements' PLY files to determine if the elements are in contact and to fill the value of the *InContact* property of the spatial relationship. In the second step, a module traverses the spatially related pairs of element nodes and tests for two conditions: contact or overlap with a pre-set tolerance, and membership of a predefined set of logically possible domain-type correspondence type pairs (such as Architectural-Floor – Structural-Slab; Architectural-Toilet – Plumbing-Toilet). If these conditions are satisfied, a *CBIM:CorrespondsTo* relationship is instantiated.

This approach does not assume that the CBIM engine establishes all spatial relationships before testing for correspondence or other design intent relationships: the modules are run repeatedly, in cycles, until no new information is added to the meta graph. In this way, each function type can reliably exploit information added by the others.



680

681 *Figure 10. Instantiated CBIM Relationships across discipline sub-graphs. The blue nodes, grey text boxes and*
682 *blue dashed lines are CBIM meta graph instances.*

683
684

Algorithm 1: Instance-level *CBIM:CorrespondsTo* relationship generation

Input: CBIM meta graph containing domain sub-graphs and *CBIM:RelSpatial* relationships across sub-graph objects: *cbim_graph*
Corresponding entity domain type pairs list: $DT = \{(dt, dt)_1, (dt, dt)_2, ...(dt, dt)_n\}$;
Predefined volume intersection tolerance threshold: ε_int;
Method adding new triples to a given graph: AddTriple();
Function returning the list of *CBIM:RelSpatial* nodes extracted from the input CBIM meta graph: ExtractRelSpatial();
Function returning the *CBIM:InContact* Boolean value of a input *CBIM:RelSpatial* node: InContact();
Function returning the *CBIM:hasSubject* node of a input *CBIM:RelSpatial* node: Subject();
Function returning the *CBIM:hasObject* node of a input *CBIM:RelSpatial* node: Object();
Function returning the ratio of volume intersection of the bounding box geometries of two input building element nodes: BBIntersect();
Function returning the ratio of volume intersection of the precise geometries of two input building element nodes: PGInersect();
Function returning the type of a input building element node: Type();

Output: Enriched CBIM meta graph with *CBIM:CorrespondsTo* relationships eablished across appropriate sub-graph objects: *cbim_graph*

```
foreach rel_spatial in ExtractRelSpatial(cbim_graph) do
    if InContact(rel_spatial) == True then
        v_i ⟵ Subject(rel_spatial)
        v_j ⟵ Object(rel_spatial)
        if {Type(v_i), Type(v_j)} ∈ DT AND BBIntersect(v_i, v_j) ≥ 1 − ε_int AND PGIntersect(v_i, v_j) ≥ 1 − ε_int then
            cbim_graph.AddTriple(v_i, "CBIM:CorrespondsTo", v_j)
        end if
    end if
end foreach
return cbim_graph
```

Figure 11. Algorithm pseudo-code for instantiating CBIM:CorrespondsTo relationships. The code itself is available in the Dataset (see Section 10).

6.4 Applications on the linked CBIM meta graph

With inter-domain relationships established and maintained among federated BIM sub-graphs, core methods from the upper CBIM logic tier (see Figure 8) can run intelligent functions on the linked CBIM database. The expected functions could include maintaining changes among multiple disciplines, object-level version control, building analyses and simulations, generative design, etc. In these paragraphs, we describe change maintenance as an example to illustrate the technical feasibility of implementing intelligent functions on linked graphs.

Changes to a building's design are made frequently in a collaborative design environment, and many, if not most, impact multiple design domains. Using existing BIM tools, designers from different disciplines need to negotiate their changes and manually modify their domain models to reflect the decisions. An automatic software that tracks, records, and propagates changes across disciplines can ease the cumbersome manual process and improve collaboration efficiency; this is the goal of the function described below.

Figure 12 describes the step-by-step process with the use case introduced in Section 3.1. The CBIM meta graph for this use case has three domain sub-graphs that are interconnected with CBIM relationships applied and maintained by the CBIM engine. In the first step (Figure 12a) an architect initiates a change in the model. When checking the model against the building code, the architect realizes that the doors are not wide enough for egress in fire emergencies, meaning they need to be widened. To make space for the new doors, the architect moves the wall 450 mm in the negative x-direction. Figure 12b shows the immediate reaction of the CBIM system in response to the user-initiated model update. The change in the x coordinate value for the wall's location is detected and the CBIM engine recomputes the spatial relationships between the wall and other building elements. Amongst others, this registers as new values for the attributes of the global x direction spatial relationship with the sprinkler pipe segment - the pipe segment is now CONTAINED_IN the wall, they are in contact, and the offset distance is 0 mm. With this update, the graph is consistent with the actual situation.

In the third step (Figure 12c), the change maintenance function checks for conformance among BIM objects: whether the established topological relationships between objects comply with the corresponding design intent constraint relationships. It reads the *CBIM:RelConstraint* that specifies the minimum requirement for a clear distance between the pipe and the wall of 20 mm, and identifies that the actual offset from the *CBIM:RelSpatial* is 0 mm. Thus, it determines that the models' current state violates the design intent. This can trigger an online notice to the designers' whose building elements participate in the relationship, which can include a recommended course of action – to move the pipe in the negative x-direction by 450 mm. Should they approve the correction, the function can execute it, as shown in Figure 12d. The change is made to the sub-graph nodes of the pipe segment and propagated back to the MEP engineer's BIM interface. Any consequential additional changes to other components of the piping system (segments connected to the pipe in question) are made locally using the internal design intent logic of the MEP BIM tool, then propagated back to the CBIM graph, which updates itself once more, and so on until all conflicts are resolved. As described above, the CBIM engine iterates continuously to identify and resolve any inconsistencies in its relationships in response to any changes. The final state of the models and their CBIM graph representations are shown in Figure 12e.

Similarly, as a *CBIM:CorrespondsTo* relationship links the architectural wall and the structural wall, the change maintenance function monitors changes to either BIM element and initiates corrections where necessary. In this case, the structural wall would be moved to coincide once more with the architectural wall, and its adjoining structural walls would be extended to heal the corners within the structural BIM tool. Moreover, the change maintenance function would also be triggered to move the sprinklers located above the doors, as their design intent relationships constrain them to be placed at the centre of the width of the doors.

To demonstrate feasibility of this function, we implemented a pipeline of functions as follows:

1) A converter written in Dynamo to communicate with a CBIM graph, including writing Revit model objects to CBIM and receiving commands from CBIM to edit the Revit model, as described in Section 6.1.
2) A function to instantiate building model elements as RDF subgraphs and save them as TTL files in the CBIM database, according to Section 6.2.
3) A rule-based algorithm to instantiate *CBIM:RelSpatial* relationships and *CBIM:CorrespondsTo* inter-domain relationships among objects across domains in accordance with the CBIM ontology determine element correspondence (as described in Section 6.3 and in Figure 11).

These functions were applied to demonstrate a use case in which an architect adjusted the height of an architectural wall in their Revit model. The change was identified and communicated to the CBIM graph. The correpsondence relationship of the architectural wall to a corresponding structural wall in the structural sub-grah was followed and the system thus notified the structural engineer, in their separate Revit client instance, of the change to the wall height, displaying the reference architectural wall on the structural model. Once the enginer corrected the height, the change was then uploaded to the sub-graph, thus completing the change maintenance process discussed in Section 6.4.

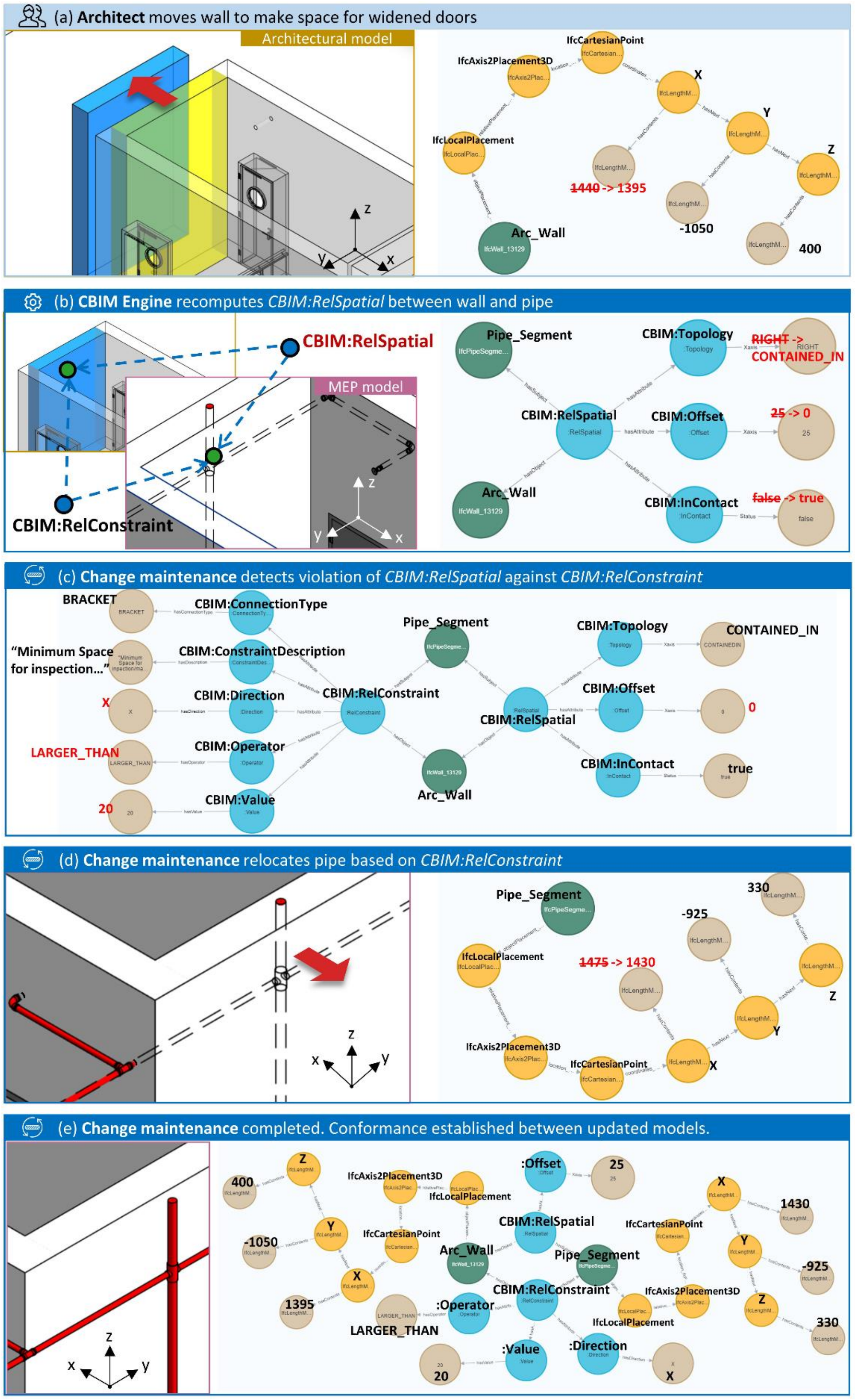


Figure 12. Change maintenance sequence for a sprinkler pipe following a move operation on an architectural wall. Screenshots of domain BIM models are shown at left and corresponding screenshots of graph database sections on the right.

7. DISCUSSION

The earlier sections of the paper have described the new paradigm, a possible system architecture, an ontology for the relationships in the meta graph, and some of the envisaged software components. In this section, we raise issues that arise with respect to the design and implementation of such a system.

7.1 Common data schema vs. native data schema

The two basic approaches to interoperability across BIM platforms to date are direct application to application translators and file export and import via an open, standard data schema (Sacks et al. 2018). The limitations of these approaches with respect to sharing parametric BIM models have restricted collaboration among design professionals because they are suited to periodic handovers of large batches of design information, not to concurrent design and engineering. Cloud storage in common data environments, governed by standards such as ISO 19650 (ISO/DIS 19650 2018), facilitates access to building information for design and construction teams but remains restricted to sharing via files. Research teams have proposed schemes for interoperability that involve using linked data with the semantic web that can provide data storage at the level of BIM objects (Pauwels et al. 2011) and global coordination across schema using a set of related ontologies (Törmä 2013) or linked data (Curry et al. 2013). CBIM proposes two advances over current practice – sharing information at the level of BIM objects and intelligent instantiation and maintenance of the topological and design intent relationships that govern the integrity of building information across domains.

However, implementation of a CBIM environment raises a central question concerning the way in which BIM authoring tools exchange information with the domain-specific sub graphs that are components of the CBIM meta graph. Here too, two possibilities arise:

1) Each BIM tool writes and reads data to a sub-graph that is defined by its own proprietary *native* schema. A CBIM platform provider could develop its system without the need for a public open schema. However, this assumes that CBIM semantic enrichment functions can be made sufficiently intelligent to recognize the BIM model objects correctly, based primarily on their location, geometry and domain identity, without relying on their naming or typing. This will depend entirely on advances in research toward semantic enrichment of BIM models (Bloch 2022).
2) Vendors of BIM tools belonging to a domain (such as structural analysis, precast concrete detailing) write and read data to a sub-graph that is defined by a domain-specific *common* schema. Specialized domain-specific open schemas could be developed on the basis of the IFC schema, with significant refinements for each design or construction discipline's domain. This would make the task of semantic enrichment for the CBIM engine's functions far simpler, as their developers would work with a known schema. However, BIM tool providers would need to prepare export and import functions that map their native schema to the common schema. Development and maintenance of the schemas would need to be governed by a public organization such as BuildingSMART.

Both are possible, but they each have advantages and disadvantages. Given the constraints described, it is likely that the first prototypes for CBIM platforms will adopt the second approach, using subsets of the current IFC schema as a temporary proxy for more sophisticated discipline specific schemas. This is the approach adopted currently for research purposes. However, should semantic enrichment become increasingly powerful, it may become possible to consider implementing platforms with the first approach.

7.2 Automation and intelligence

Section 6.3 detailed the functionality required in the CBIM engine for generating design intent relationships between building element nodes belonging to different sub-graphs, and illustrated it with a rule-based algorithm for inferring a correspondence relationship. Rule-based inferencing of this kind works by encapsulating experts' knowledge and experience as IF-THEN rules, and it is suitable for relationships that can be abstracted as machine-readable logic statements. However, some of the design intent logical relationships do not lend themselves to straightforward expression of this kind. Bloch et al. (2020) explored a range of semantic enrichment tasks and concluded that using machine-learning techniques to discover the latent feature patterns from datasets were suitable for identification and classification of fuzzy inter-domain relationships. A benefit of storing BIM models as graphs is that the format can explicitly represent the relationships among BIM entities as another dimension's features, resulting in richer information inside each BIM graph. Moreover, storing data in graphs enables use of a set of novel machine learning techniques, graph neural networks (GNN). GNNs apply deep learning operators on graphs directly, and they can consider both non-graph features and relationship-related features. Experiments have shown that GNNs perform better than other non-graph machine learning algorithms in the task of classifying BIM objects and concepts in models (Wang et al. 2021a). The application of GNNs to generate fuzzy inter-domain relationships in the CBIM engine is a valuable future research direction.

Given the need to identify a diverse range of design intent relationships, a CBIM engine will likely apply a combination of different types of algorithms, including rule-based inferencing, machine learning, deep learning, GNN, and others. The engine's semantic enrichment functions would be run recursively on the meta graph, concluding any given run only if the last full cycle had not resulted in any change to the data. The length of such cycles will be a key determinant in the feasibility of the CBIM system as a whole, as users will expect almost real-time response to changes they make. Thus, some consideration will need to be given, in research and in development, to optimal sequencing of the functions for minimal overall model update. Parallel operations on large graphs will likely be necessary.

Machine-learning will require large datasets containing the meta graphs of large numbers of building projects. Such a data set can be expected to accumulate over time and to be available to the CBIM service provider. A data set of this kind would serve not only for gradual improvement of the CBIM engine's sematic enrichment functions – it could also support generation of a wide range of intelligent functions such as data-centric design generation, checking model quality, checking conformance to building design codes, and others. Given the presence of enriched semantic information in the meta graphs, this is more likely to be successful than current research attempts to learn from BIM models (Zabin et al. 2022).

If a native schema approach (see the previous subsection, 7.1) were adopted, machine learning could also be applied to enhance the functionality of the CBIM functions by learning the proprietary native schemas of the sub-graphs written to the system by external BIM tools. Indeed, proprietary schemas can be regarded as data structures, and learning the latent data structure distribution from given samples is a typical task of unsupervised learning. This approach, relying on unsupervised learning, requires big data sets.

Similarly, very large data sets of multi-disciplinary building models may afford the opportunity for unsupervised learning to discover design intent relationships that were not identified initially. Given the complexity and depth of graph relationship chains that would be needed, and the limitations of current state-of-the-art unsupervised learning algorithms designed for graphs, this should be seen as a long-term capability.

7.3 Core layer and extension layer

Thus far, we have explored and explained the CBIM graph subject to the assumption that all the information describing a building is present in the domain sub-graphs. However, from experiments with the full-scale federated building model described in Chapter 3, it is apparent that RDF graphs compiled from IFC data files with full geometry are too complex for practical engineering applications (Pauwels and Roxin 2017). The depth of the product representation chains renders the graphs opaque to machine-learning techniques in general, and to GNNs in particular. A simpler, yet full representation of the models is desirable.

The BOT ontology, which focuses on the overall spatial structure of a building, is an attempt to represent a building in linked data in a way that is useful for queries and learning (Rasmussen et al. 2021). However, it excludes much of the detailed information needed for a CBIM system. Recognizing the nature of 3D solid geometry data, and the fact that the semantic relationships inferred by CBIM are established at the building element level rather than at the property level, it would be practical to separate all detailed geometry and selected alphanumeric property sets to separate files associated with building element nodes in the graph. The meta graph and sub-graphs form the *core layer*, solid geometry files and property set tables form the *extension layer*. This would maximise semantic expressiveness in the core layer for processing by AI algorithms, while geometry and property sets are accessible in formats that are better suited to their content and manipulation. This framework for model representation coincides with the efforts of the W3C Linked Building Data Community Group in devising a modularized and extensible graph representation for BIM models. Their IFCtoLBD converter (Oraskari et al. 2021) allows IFC models to be converted into RDF Abox graphs based on modular ontologies, with core building information preserved and complex properties such as geometries omitted. Furthermore, Pauwels et al. (2022) have recommended ways to link external geometry files to RDF components, including use of *bot:hasSimple3DModel* or *omg:hasSimpleGeometryDescription* relationships. This approach was adopted here, with exact object geometries stored as external PLY format files.

Knowledge of the topology relationships between building elements is central to many of the semantic enrichment tasks within the CBIM engine and for intelligent applications that will work on the platform. CBIM resolves this issue by making the relationships explicit in the meta graph, immediately available for algorithms and for learning by GNNs. In the current implementation, close-fit bounding box geometry and position and orientation data are stored explicitly in the graphs, thus allowing 'first pass' topology checking routines – such as the $\text{BBIntersect}(v_i, v_j)$ method of Figure 11 – to run directly within the graph without the need to access and process information from the extension layer.

7.4 Limitations and future research

The nature of research in the application of linked data technologies to BIM applications to date has been exploratory. It has not sought to disprove any hypothesis, but rather to explore the boundaries of the feasible. This work contributes to this thread of work by outlining those boundaries, describing a system solution that purports to fulfil the needs for concurrent, collaborative multi-disciplinary design and construction. The experimental implementation demonstrates feasibility of some of the functionality described for the CBIM paradigm, but not all that is required. Such proof of feasibility will only be achieved when researchers implement all the key components and demonstrate that they are individually feasible.

The research is also constrained by practical limitations. The first is the absence of user-interfacing design tools that can manipulate graphs directly: all current commercial BIM authoring tools work with proprietary file storage. In the experiments we used these tools to model buildings and translated the building models into graph formats (using IFCtoRDF or IFCtoLBD (Oraskari et al. 2021; Pauwels 2021)) and by implementing Dynamo scripts for direct translation of BIM models to graphs. Secondly, there is no extant listing or classification of design intent relationships. While these are well understood by practitioners, most are implicit knowledge, and they must be elicited by researchers. A set of explicit design intent relationships will be needed before efforts to prove the feasibility of identifying them automatically using AI techniques can be attempted. Thirdly, there are currently no analysis or simulation tools that can leverage a CBIM type meta graph to provide their input – researchers will need to develop prototypes of tools of this type too before they can begin to flesh out the potential benefits of a CBIM system. Finally, once some of these components are in place, or can be simulated, research will be needed to establish the business case potential, which itself is dependent on design research to establish the feasibility of truly concurrent detailed design across building design disciplines.

Future research will be needed to address issues such as processing efficiency of the CBIM engine in compiling spatial relationships and in semantic enrichment to express design intent. Instantiating spatial relationships requires geometry analysis which may be slow for large models if done in batches – incremental upload to the cloud will be preferable. Semantic enrichment may reveal conflicting design constraints, which are common in building design – instantiating such constraints will be helpful in making them explicit to designers, but attempting to process them may lead to cycles of changes. Change management will require research to determine optimal sequences of diagnosis and propagation with the goal of minimizing response latency for users, such as those proposed by Hu et al. (2020).

Progress toward realization of CBIM services will also require extensive research on topics such as:

- Domain-specific ontologies for building design, fabrication, and construction,
- Graph learning algorithms for BIM data, methods to apply them for semantic enrichment, and suitable graph representations,
- Alternative hybrid storage strategies to optimize the core layer and extension layer technologies, and
- Extension and refinement of the CBIM ontology to represent newly identified design intent relationships.

8. CONCLUSION

This paper proposes and explicates a new paradigm for BIM, called 'CBIM', in which federated domain-specific BIM models are represented as property graphs on a cloud platform with relationships that store design intent and spatial relationships among building objects explicitly. The purpose of CBIM is to facilitate concurrent collaboration among designs, automating much of the leg work currently required to maintain consistency across domain models. A functioning system would allow designers to work within their domain-specific BIM tools while actions to maintain model consistency and design intent are performed in the background within the cloud database itself.

Specifically, the work described contributes a) specification of the new paradigm, b) a design for the system architecture, c) an ontology that details the inter-domain for spatial, topological and design intent constraint relationships, d) explanation of the function of the various software components, and e) demonstration of the feasibility of supplementing spatial and topological relationships in a meta graph, f) demonstration of the feasibility of maintain inter-domain model consistency following design changes, and g) discussion of the technical, professional, and commercial considerations and limitations.

The CBIM paradigm represents a significant break not only from the current state of the art in commercial BIM software and BIM processes, but also from the current research trends in BIM interoperability. It offers a route to implementation of a comprehensive BIM system by a commercial platform enterprise which would allow vendors of discipline-specific BIM tools to integrate their applications with the central data store, as well as enabling vendors of building simulation and analysis software tools to operate directly on a building model in the cloud that contained comprehensive, multi-disciplinary information. It also presents extensive opportunities for future development of intelligent automated design agents, including design checking and code compliance tools. However, CBIM requires integration through application of automated, intelligent semantic enrichment to identify and apply its inter-domain relationships. While experiments conducted in this and in other efforts have demonstrated feasibilty for simple situations with rule inferencing, much research is still needed.

9. ACKNOWLEDGEMENT

This research was supported in part by the CBIM project, which received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860555.

10. DATASET

The CBIM ontology, BIM model files, graph database files, SPARQL code and instructions for validation and replication of the change management pipeline can be downloaded from a GitHub repository at

https://github.com/terry-oy/CBIM-position

11. REFERENCES

Abdul-Ghafour, S., P. Ghodous, B. Shariat, and E. Perna. 2007. "A Common Design-Features Ontology for Product Data Semantics Interoperability." *Proc. IEEEWICACM Int. Conf. Web Intell. WI 2007*, 427–430. IEEE.

Abdul-Ghafour, S., P. Ghodous, B. Shariat, and E. Perna. 2011. "Ontology development for the integration of CAD models in a collaborative environment." *Adv. Concurr. Eng.*, 207–214. https://doi.org/10.1007/978-0-85729-799-0_24.

Abdul-Ghafour, S., P. Ghodous, B. Shariat, E. Perna, and F. Khosrowshahi. 2014. "Semantic interoperability of knowledge in feature-based CAD models." *CAD Comput. Aided Des.*, 56: 45–57. Elsevier. https://doi.org/10.1016/j.cad.2014.06.001.

Acemoglu, D., U. Akcigit, and M. A. Celik. 2020. "Radical and incremental innovation: The roles of firms, managers and innovators." *AEJ Macroecon Forthcom.*

Afsari, K., C. Eastman, and D. Shelden. 2017. "Building Information Modeling data interoperability for Cloud-based collaboration: Limitations and opportunities." *Int. J. Archit. Comput.*, 15 (3): 187–202. SAGE Publications. https://doi.org/10.1177/1478077117731174.

Akintoye, A., J. Goulding, and G. Zawdie (Eds.). 2012. *Construction innovation and process improvement*. Chichester, West Sussex, U.K. ; Ames, Iowa: Wiley-Blackwell.

Aljohani, A. 2017. "Construction Projects Cost Overrun: What Does the Literature Tell Us?" *Int. J. Innov. Manag. Technol.*, 137–143. https://doi.org/10.18178/ijimt.2017.8.2.717.

Anumba, C. J., and N. F. O. Evbuomwan. 1997. "Concurrent engineering in design-build projects." *Constr. Manag. Econ.*, 15 (3): 271–281. Routledge. https://doi.org/10.1080/014461997373006.

Anumba, C. J., J. M. Kamara, and A.-F. Cutting-Decelle (Eds.). 2007. *Concurrent engineering in construction projects*. London ; New York: Taylor & Francis.

Balaji, B., A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Berges, D. Culler, R. Gupta, M. B. Kjærgaard, M. Srivastava, and K. Whitehouse. 2016. "Brick: Towards a Unified Metadata Schema For Buildings." *Proc. 3rd ACM Int. Conf. Syst. Energy-Effic. Built Environ.*, 41–50. Palo Alto CA USA: ACM.

Ballard, G. 2000. "Positive vs negative iteration in design." *Proc. Eighth Annu. Conf. Int. Group Lean Constr.*, 17–19. Brighton, UK: IGLC.

Bloch, T. 2022. "Connecting research on semantic enrichment of BIM - review of approaches, methods and possible applications." *J. Inf. Technol. Constr.*, 27: 416–440. https://doi.org/10.36680/j.itcon.2022.020.

Bloch, T., and R. Sacks. 2020. "Clustering Information Types for Semantic Enrichment of Building Information Models to Support Automated Code Compliance Checking." *J. Comput. Civ. Eng.*, 34 (6): 04020040. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000922.

Bonduel, M., J. Oraskari, P. Pauwels, M. Vergauwen, and R. Klein. 2018. "The IFC to linked building data converter - Current status." *CEUR Workshop Proc.*, 34–43.

Bonino, D., and F. Corno. 2008. "DogOnt - Ontology Modeling for Intelligent Domotic Environments." *Semantic Web - ISWC 2008*, Lecture Notes in Computer Science, A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, eds., 790–803. Berlin, Heidelberg: Springer.

Chandrasekaran, B., and J. R. Josephson. 2000. "Function in Device Representation." *Eng. Comput.*, 16 (3): 162–177. https://doi.org/10.1007/s003660070003.

Curry, E., J. O'Donnell, E. Corry, S. Hasan, M. Keane, and S. O'Riain. 2013. "Linking building data in the cloud: Integrating cross-domain building data using linked data." *Adv. Eng. Inform.*, 27 (2): 206–219. https://doi.org/10.1016/j.aei.2012.10.003.

Dalpadulo, E., F. Pini, and F. Leali. 2020. "Integrated CAD platform approach for Design for Additive Manufacturing of high performance automotive components." *Int. J. Interact. Des. Manuf. IJIDeM*, 14 (3): 899–909. https://doi.org/10.1007/s12008-020-00684-7.

Daniele, L., R. Garcia-Castro, M. Lefrançois, and M. Poveda-Villalon. 2022. "SAREF: the Smart Applications REFerence ontology." Accessed January 10, 2022. https://saref.etsi.org/core/.

Domingue, J., D. Fensel, and J. A. Hendler (Eds.). 2011. *Handbook of Semantic Web Technologies*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Egenhofer, M. J. 1989. "A Formal Definition of Binary Topological Relationships." *Proc. 3rd Int. Conf. Found. Data Organ. Algorithms*, FOFO '89, 457–472. Berlin, Heidelberg: Springer-Verlag.

Escrig-Tena, A. B., M. Segarra-Ciprés, and B. García-Juan. 2021. "Incremental and radical product innovation capabilities in a quality management context: Exploring the moderating effects of control mechanisms." *Int. J. Prod. Econ.*, 232: 107994. https://doi.org/10.1016/j.ijpe.2020.107994.

Esser, S., S. Vilgertshofer, and A. Borrmann. 2021. "Graph-based version control for asynchronous BIM level 3 collaboration." *EG-ICE 2021 Workshop Intell. Comput. Eng.*, J. Abualdenien, A. Borrmann, L. C. Ungureanu, and T. Hartmann, eds., 98–107. Berlin: Technische Universität Berlin.

Fischer, M., A. Khanzode, D. Reed, and H. W. Ashcraft. 2017. *Integrating Project Delivery*. Wiley.

Franzke, M., E. Pilke, and J. Bacus. 2021. *Eidos User Research Initial Readout*. Unpublished internal document. Espoo, Finland: Trimble Inc.

Ghassemi, R., and B. Becerik-Gerber. 2011. “Transitioning to Integrated Project Delivery: Potential barriers and lessons learned.” *Lessons Learn.*, 21.

GraphDB. 2021. “GraphDB Semantic Graph Database.” Accessed September 21, 2021. https://graphdb.ontotext.com/.

Guide, A. I. A. 2007. “Integrated project delivery: A guide.” *Am. Inst. Archit. Calif.*

Hu, Y., D. Castro-Lacouture, C. M. Eastman, and S. B. Navathe. 2020. “Automatic clash correction sequence optimization using a clash dependency network.” *Autom. Constr.*, 115: 103205. https://doi.org/10.1016/j.autcon.2020.103205.

IFC. 2018. “IFC4 Add2 Release.” Accessed November 6, 2018. http://www.buildingsmart-tech.org/specifications/ifc-releases/ifc4-add2.

Isaac, S. 2011. “Development of a Model for Evaluating the Impact of Proposed Changes in Construction Projects.” PhD Thesis. Haifa: Technion-Israel Institute of Technology, Faculty of Civil and Environmental Engineering.

Ismail, A., A. Nahar, and R. Scherer. 2017. “Building knowledge extraction from BIM/IFC data for analysis in graph databases.” *Digit. Proc. 24th EG-ICE Int. Workshop Intell. Comput. Eng. 2017*, 146–157.

ISO/DIS 19650. 2018. “Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) -- Information management using building information modelling -- Part 1: Concepts and principles.”

Kent, D. C., and B. Becerik-Gerber. 2010. “Understanding Construction Industry Experience and Attitudes toward Integrated Project Delivery.” *J. Constr. Eng. Manag.*, 136 (8): 815–825. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000188.

Koo, B., R. Jung, and Y. Yu. 2021a. “Automatic classification of wall and door BIM element subtypes using 3D geometric deep neural networks.” *Adv. Eng. Inform.*, 47: 101200. Elsevier. https://doi.org/10.1016/j.aei.2020.101200.

Koo, B., R. Jung, Y. Yu, and I. Kim. 2021b. “A geometric deep learning approach for checking element-to-entity mappings in infrastructure building information models.” *J. Comput. Des. Eng.*, 8 (1): 239–250. https://doi.org/10.1093/jcde/qwaa075.

Koo, B., S. La, N. W. Cho, and Y. Yu. 2019. “Using support vector machines to classify building elements for checking the semantic integrity of building information models.” *Autom. Constr.*, 98: 183–194. Elsevier B.V. https://doi.org/10.1016/j.autcon.2018.11.015.

Koo, B., and B. Shin. 2018. “Applying novelty detection to identify model element to IFC class misclassifications on architectural and infrastructure Building Information Models.” *J. Comput. Des. Eng.*, 5 (4): 391–400. Society for Computational Design and Engineering. https://doi.org/10.1016/j.jcde.2018.03.002.

Krech, D. 2013. “rdflib Documentation.” Accessed November 21, 2021. https://rdflib.readthedocs.io/en/stable/#in-depth.

Lai, H., and X. Deng. 2018. “Interoperability analysis of IFC-based data exchange between heterogeneous BIM software.” *J. Civ. Eng. Manag.*, 24 (7): 537–555. https://doi.org/10.3846/jcem.2018.6132.

Le, T., and H. D. Jeong. 2016. “Interlinking life-cycle data spaces to support decision making in highway asset management.” *Autom. Constr.*, 64: 54–64. Elsevier. https://doi.org/10.1016/j.autcon.2015.12.016.

Lee, G., R. Sacks, and C. M. Eastman. 2006. “Specifying Parametric Building Object Behavior (BOB) for a Building Information Modeling System.” *Autom. Constr.*, 15: 758–776.

Ma, L., R. Sacks, U. Kattel, and T. Bloch. 2018. "3D Object Classification Using Geometric Features and Pairwise Relationships." *Comput.-Aided Civ. Infrastruct. Eng.*, 33 (2): 152–164. Blackwell Publishing Inc. https://doi.org/10.1111/mice.12336.

Manola, F., E. Miller, B. McBride, and others. 2004. "RDF primer." *W3C Recomm.*, 10 (1–107): 6.

Mattern, H., and M. König. 2018. "BIM-based modeling and management of design options at early planning phases." *Adv. Eng. Inform.*, 38: 316–329. https://doi.org/10.1016/j.aei.2018.08.007.

Matthews, J., P. E. D. Love, J. Mewburn, C. Stobaus, and C. Ramanayaka. 2018. "Building information modelling in construction: insights from collaboration and change management perspectives." *Prod. Plan. Control*, 29 (3): 202–216. Taylor & Francis. https://doi.org/10.1080/09537287.2017.1407005.

Migliari, R. 2012. "Descriptive Geometry: From its Past to its Future." *Nexus Netw. J.*, 14 (3): 555–571. https://doi.org/10.1007/s00004-012-0127-3.

Mohd Nawi, M. N., N. H. Baluch, and A. Y. Bahaudin. 2014. "Impact of fragmentation issue in construction industry: An overview." *MATEC Web Conf.*, 15: 01009. EDP Sciences.

Nour, M., and K. Beucke. 2010. "Object versioning as a basis for design change management within a BIM context." *Proc. 13th Int. Conf. Comput. Civ. Build. Eng. ICCCBE-XIII Nottm. UK*.

Onshape. 2022. "Onshape | Product Development Platform." Accessed May 29, 2022. https://www.onshape.com/en/.

Oraee, M., M. R. Hosseini, D. J. Edwards, H. Li, E. Papadonikolaki, and D. Cao. 2019. "Collaboration barriers in BIM-based construction networks: A conceptual model." *Int. J. Proj. Manag.*, 37 (6): 839–854. https://doi.org/10.1016/j.ijproman.2019.05.004.

Oraee, M., M. R. Hosseini, E. Papadonikolaki, R. Palliyaguru, and M. Arashpour. 2017. "Collaboration in BIM-based construction networks: A bibliometric-qualitative literature review." *Int. J. Proj. Manag.*, Social Responsibilities for the Management of Megaprojects, 35 (7): 1288–1301. https://doi.org/10.1016/j.ijproman.2017.07.001.

Oraskari, J., K. McGlinn, P. Pauwels, F. Priyatna, A. Wagner, and J. L. OH64K. 2021. *jyrkioraskari/IFCtoLBD: IFCtoLBD 2.33.0*. Zenodo.

Pauwels, P. 2021. "IFCtoRDF." *GitHub*. Accessed September 21, 2021. https://github.com/pipauwel/IFCtoRDF.

Pauwels, P., A. Costin, and M. H. Rasmussen. 2022. "Knowledge Graphs and Linked Data for the Built Environment." *Ind. 40 Built Environ. Methodol. Technol. Ski.*, M. Bolpagni, R. Gavina, and D. Ribeiro, eds., 157–183. Cham: Springer International Publishing.

Pauwels, P., R. De Meyer, and J. Van Campenhout. 2011. "Interoperability for the Design and Construction Industry through Semantic Web Technology." *Semantic Multimed.*, Lecture Notes in Computer Science, T. Declerck, M. Granitzer, M. Grzegorzek, M. Romanelli, S. Rüger, and M. Sintek, eds., 143–158. Berlin, Heidelberg: Springer.

Pauwels, P., and A. Roxin. 2017. "SimpleBIM: From full ifcOWL graphs to simplified building graphs." *EWork EBusiness Archit. Eng. Constr.*, 11–18. CRC Press.

Pauwels, P., and W. Terkaj. 2016. "EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology." *Autom. Constr.*, 63: 100–133. Elsevier. https://doi.org/10.1016/j.autcon.2015.12.003.

Pilehchian, B., S. Staub-French, and M. P. Nepal. 2015. "A conceptual approach to track design changes within a multi-disciplinary building information modeling environment." *Can. J. Civ. Eng.*, 42 (2): 139–152. https://doi.org/10.1139/cjce-2014-0078.

Rasmussen, M. H., M. Lefrançois, G. F. Schneider, and P. Pauwels. 2021. "BOT: The building topology ontology of the W3C linked building data group." *Semantic Web*, 12 (1): 143–161. IOS Press. https://doi.org/10.3233/SW-200385.

Reinisch, C., M. J. Kofler, and W. Kastner. 2010. "ThinkHome: A smart home as digital ecosystem." *4th IEEE Int. Conf. Digit. Ecosyst. Technol.*, 256–261.

Rezgui, Y., A. Brown, G. Cooper, P. Brandon, and M. Betts. 1998. "Intelligent models for collaborative construction engineering." *Comput.-Aided Civ. Infrastruct. Eng.*, 13 (3): 151–161. https://doi.org/10/b3rfvz.

Sacks, R., C. M. Eastman, G. Lee, and P. Teicholz. 2018. *BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors and Facility Managers*. Hoboken, NJ: John Wiley and Sons.

Sacks, R., L. Ma, R. Yosef, A. Borrmann, S. Daum, and U. Kattel. 2017. "Semantic Enrichment for Building Information Modeling: Procedure for Compiling Inference Rules and Operators for Complex Geometry." *J. Comput. Civ. Eng.*, 31 (6): 04017062. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000705.

Safdari, F. 2018. "Concurrent Engineering in Construction Projects - Lesson learned from the oil and gas industry."

Segaran, T., J. Taylor, and C. Evans. 2009. *Programming the Semantic Web*. Beijing ; Sebastopol, CA: O'Reilly.

Sierra-Aparicio, M. V., J. L. Ponz-Tienda, and J. P. Romero-Cortés. 2019. "BIM Coordination Oriented to Facility Management." *Adv. Inform. Comput. Civ. Constr. Eng.*, I. Mutis and T. Hartmann, eds., 123–128. Cham: Springer International Publishing.

Spatial. 2021. *3D ACIS Modeler*. Broomfield, Colorado, USA: Spatial Corp.

Stjepandić, J., N. Wognum, and W. J.C. Verhagen (Eds.). 2015. *Concurrent Engineering in the 21st Century*. Cham: Springer International Publishing.

Studer, R., V. R. Benjamins, and D. Fensel. 1998. "Knowledge engineering: Principles and methods." *Data Knowl. Eng.*, 25 (1–2): 161–197. https://doi.org/10.1016/S0169-023X(97)00056-6.

Törmä, S. 2013. "Semantic linking of building information models." *Proc. - 2013 IEEE 7th Int. Conf. Semantic Comput. ICSC 2013*, 412–419.

Wagner, A., M. Bonduel, and P. Pauwels. 2022. "OMG: Ontology for Managing Geometry." Accessed January 11, 2022. https://www.projekt-scope.de/ontologies/omg/#.

Wang, Z., R. Sacks, and T. Yeung. 2021a. "Exploring graph neural networks for semantic enrichment: Room type classification." *Autom. Constr.*, 104039. https://doi.org/10.1016/j.autcon.2021.104039.

Wang, Z., Yeung, T., Sacks, R. and Su, Z. (2021), "Room Type Classification for Semantic Enrichment of Building Information Modeling Using Graph Neural Networks", Proceedings of the 38th International Conference of CIB W78, Luxembourg, 13-15 October, pp 773-781 (ISSN: 2706-6568), http://itc.scix.net/paper/w78-2021-paper-077.

Wu, D., D. W. Rosen, L. Wang, and D. Schaefer. 2015. "Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation." *Comput.-Aided Des.*, 59: 1–14. https://doi.org/10.1016/j.cad.2014.07.006.

Wu, D., J. Terpenny, and D. Schaefer. 2017. "Digital design and manufacturing on the cloud: A review of software and services—RETRACTED." *Artif. Intell. Eng. Des. Anal. Manuf.*, 31 (1): 104–118. https://doi.org/10.1017/S0890060416000305.

Yang, Q. Z., and Y. Zhang. 2006. "Semantic interoperability in building design: Methods and tools." *CAD Comput. Aided Des.*, 38 (10): 1099–1112. Elsevier. https://doi.org/10.1016/j.cad.2006.06.003.

Yassine, A., and D. Braha. 2003. "Complex Concurrent Engineering and the Design Structure Matrix Method." *Concurr. Eng.*, 11 (3): 165–176. SAGE Publications Ltd STM. https://doi.org/10.1177/106329303034503.

Yazdani, B. 1999. “Four Models of Design Definition: Sequential, Design Centered, Concurrent and Dynamic.” *J. Eng. Des. - J Eng. Des.*, 10: 25–37. https://doi.org/10.1080/095448299261407.
Yoo, S., S. Lee, S. Kim, K. H. Hwang, J. H. Park, and N. Kang. 2021. “Integrating deep learning into CAD/CAE system: generative design and evaluation of 3D conceptual wheel.” *Struct. Multidiscip. Optim.*, 64 (4): 2725–2747. https://doi.org/10.1007/s00158-021-02953-9.
Zabin, A., V. A. González, Y. Zou, and R. Amor. 2022. “Applications of machine learning to BIM: A systematic literature review.” *Adv. Eng. Inform.*, 51: 101474. https://doi.org/10.1016/j.aei.2021.101474.
Zada, A. J., W. Tizani, and A. H. Oti. 2014. “Building Information Modelling (BIM)—Versioning for Collaborative Design.” 512–519. American Society of Civil Engineers. https://doi.org/10.1061/9780784413616.064.
Zhao, X., Y. Feng, J. Pienaar, and D. O’Brien. 2017. “Modelling paths of risks associated with BIM implementation in architectural, engineering and construction projects.” *Archit. Sci. Rev.*, 60 (6): 472–482. Taylor & Francis. https://doi.org/10.1080/00038628.2017.1373628.
Zhou, Q., S.-K. Choi, and R. M. Gorguluarslan. 2020. “Editorial for the Special Issue: Computer-Aided Manufacturing and Design.” *Appl. Sci.*, 10 (16): 5650. https://doi.org/10.3390/app10165650.

Accepted for Advanced Engineering Informatics, June 2022 https://doi.org/10.1016/j.aei.2022.101711