Automation in Construction 173 (2025) 106081

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

BIM-based semantic enrichment and knowledge graph generation via geometric relation checking

Georgios Nektarios Lilis *, Meng Wang, Kyriakos Katsigarakis, Dimitrios Mavrokapnidis, Ivan Korolija, Rovas Dimitrios

University College London, London, United Kingdom

ARTICLE INFO

Keywords:
BIM
IFC
Geometry
Interoperability
Digital twins
Semantic web technologies

ABSTRACT

Building Information Models support information exchange and collaboration between designers, engineers and stakeholders of the built environment. Due to the large scale and multi-domain requirements of building projects, building information is often fragmented to multiple files, prone to modeling errors and poor level of detail. As a result, BIM data cannot be reused and remain siloed across the building lifecycle. This paper introduces the Geometric Relation Checking tool, a novel tool that automatically detects geometric relations between IFC objects. These relationships can be used to infer missing semantic relationships among these objects, increasing the inter-connectivity among semantic graphs of different domains and at the same time breaking their siloed structures. The tool is tested on MEP and architecture domain data, but its use can be generalized to any other data domain that contains elements with geometric representations.

1. Introduction

Building Information Modeling (BIM) has transformed the construction industry by creating a collaborative and comprehensive data environment that supports building design, construction, and operation. BIM models contain detailed information – including geometry, spatial relationships, and semantic data – that enables the analysis and simulation of various aspects of the building, such as structural behavior, energy performance, and construction sequencing. An open data format is essential to facilitate seamless data exchange among stakeholders who use different software applications. Industry Foundation Classes (IFC) provide this interoperability as an open standard-based file format, making it a cornerstone of the BIM process by enabling BIM data to be shared across diverse platforms.

Ensuring accurate and consistent geometric relationships between building elements, or the spatial information that defines how pairs of elements relate (relative position, proximity, containment, …), is essential for effectively implementing BIM-based workflows in an IFC model. While tools for checking geometric relationships exist to enhance the quality of IFC models (Solibri, TEKLA model checker, and others), the potential to leverage these relationships using spatial reasoning for inferring non-geometric semantics remains relatively unexplored. For example, spatial relationships between elements – such as adjacency, containment, and intersection – can help infer non-geometric relationships. Adjacency might indicate connectivity when one element is attached to another, while containment could imply inheritance when one element is enclosed within another. Manually checking these relationships by visual inspection is a time-consuming and error-prone task, particularly in large and complex models. Therefore, automated geometry relation-checking processes are essential to help BIM users detect these relationships and infer other non-geometric semantics.

The need for multi-domain collaborative information exchange and less rigid information exchange structures appears to be driving increasing interest in the use of property and knowledge graphs [1]. Such approaches are increasingly useful in operations and management, where several ontologies have been introduced and standards such as ASHRAE 223P [2] are emerging [3]. The understanding is that in many such cases, detailed geometric information is not important, but rather topological relations among elements suffice. Yet extracting such topological information remains a manual – and often tedious – task that requires an understanding of the geometric relations between building elements to infer topological information. Extracting the topological information is a straightforward task in an ideal case where all relationships have been explicitly and correctly modeled. Yet, enforcing such strict quality requirements on the BIM models can be tedious and unrealistic and has been shown in practice to be hard to attain—even when the best intentions exist. The reality is that available models can have modeling omissions or errors, which are compounded

* Corresponding author.
E-mail address: g.lilis@ucl.ac.uk (G.N. Lilis).

https://doi.org/10.1016/j.autcon.2025.106081
Received 13 August 2024; Received in revised form 14 February 2025; Accepted 19 February 2025

by imperfect information transfer between authoring tools and, more broadly, issues related to information extraction and transfer. Being able to use the geometric context from a BIM model, even when such a model is imperfect or incomplete, to infer topological information, can be quite useful. Extracting such topological information can be useful in several contexts: for example, in the architectural context to infer connectivity between spaces, or in the Mechanical Electrical and Plumbing (MEP) context to establish fluid flows (air, water, or refrigerant) through distribution networks. Regardless of the domain, addressing the challenge of extracting topological information requires returning to the geometric space. The use of geometric relations among elements can help uncover additional semantic relations, regardless of the domains, provided that a sufficient geometric context exists in these domains.

Motivated by the above, we introduce a four-step process that involves: the operations of a novel Geometric Relation Checking (GRC) tool at its core and the use of a dedicated Geometry Exporter [4], for the extraction of the IFC geometric context and its transformation to a GRC-compatible input.

The primary objective of this paper is to introduce a methodology that uses the previous four-step process, to extract topological information from geometric data and generate or enrich a knowledge graph with this information. To demonstrate and evaluate the proposed methodology, we consider a specific and challenging use case: extracting the Heating, Ventilation, and Air Conditioning (HVAC) topology of a building in a knowledge graph form, for use in operations and maintenance (O&M) contexts or energy performance simulations. For this purpose, we need to use the solid geometric representations of Mechanical Electrical and Plumbing (MEP) components and building spaces, as input. We apply the GRC process on these representations, to identify the geometric relationships among these components and, based on these relationships, infer their connectivity patterns and the locations of the HVAC terminal units, to instantiate the final HVAC topology knowledge graph.

Considering the above, the structure of the paper evolves as follows: first, the background section provides a review on Geometric Relation Checking (GRC) concepts. The four-step process is detailed in Section 2. Section 4 demonstrates the application of the introduced process to MEP fluid transfer networks, followed by examples of this application in Section 5. The paper concludes with three final sections: Section 6 discusses the capabilities of the process, Section 7 addresses its limitations and potential future improvements, and Section 8 resumes the work.

2. Background

Geometric relationships between the geometric representations of solid BIM elements can be used to infer relationships like adjacency, clash, or containment, and it is broadly used when working with BIM and related applications. We call the generic ability to detect such relationships, Geometric Relation Checking (GRC); the concept of GRC is quite general and can have broad and far-reaching applications. Typical uses include (i) clash detection to ensure a BIM model meets threshold quality requirements, (ii) compliance checking, (iii) or for semantic enrichment of BIM models and the generation of knowledge graphs. We discuss each of these areas below to highlight potential uses of our novel GRC tool that performs GRC operations. In this paper, we focus on the ability to infer topological relationships using GRC, which can be quite useful in several contexts that have not been adequately explored in the literature.

2.1. Computer-aided design

The concept of geometric relation checking (GRC) emerged in an early form within computer-aided design (CAD). While the notion of a construction or building element was absent in initial CAD data, geometric relations such as distances and constraints among object lines in two dimensions or object faces in three dimensions were evaluated through a process called dimensioning [5]. Dimensioning involves adding measurements – such as length, width, and height – to a design, which serve to verify the accuracy of geometric relationships, including distances and angles relative to an axis. Constraints, on the other hand, are rules defining relationships between elements in CAD, such as parallelism or perpendicularity. These fundamental GRC functions helped ensure design accuracy and manufacturability in early CAD. At a later stage, five specific types of geometric relations were introduced in 3D architectural CAD data: adjacency, containment, intersection, separation, and connectivity [6].

2.2. Building information models

In transitioning from CAD to Building Information Modeling (BIM), the concept of a building or construction element – often referred to as a BIM element in a broad sense – became more explicit. In the BIM context, GRC is used to verify relationships among the geometric representations of BIM elements. These checks play a crucial role in automated rule-checking mechanisms [7], facilitate 3D information retrieval by exploring model topology [8], and support object classification processes [9,10]. In this context, GRC can be performed using the solid representations of octrees [11]. Octrees are geometric data structures used to approximate solid objects by recursively dividing 3D space into eight smaller cubic regions, or octants until each reaches a defined minimum volume. Each node in an octree contains eight child nodes. This structure enables GRC operations between pairs of objects by executing these operations on their corresponding child nodes. Finally, the concept of geometric relation checking among architectural BIM elements has been used for graph data model creation [12] and dimension reduction using graph models for simulation purposes [13,14].

2.3. Clash detection

The geometric relation, defined as clash, when two solid objects have a non-trivial intersection, has been extensively used as a BIM quality-checking mechanism. Clash detection [15] is used to ensure that elements from different domains do not intersect. For instance, it verifies that architectural elements, such as walls and floors, do not collide with MEP elements, including mechanical, electrical, and plumbing systems. Also, clash detection has been used to support cross-domain BIM coordination problems [16,17] and reduce costs that occur from design errors and potential construction errors [18]. Clashes were classified further as geometric relations in Hu et al. [19]. Clashes are significant as incorrect or incomplete geometric representations intersecting in a general 4D-BIM setup can lead to design errors, construction delays, increased costs, and health and safety issues [20].

2.4. Code compliance checking

Another area where the GRC can provide insights is the automatic code compliance checking [21]. Within this subject, certain checking rules can be interpreted as relations among the geometric representations of BIM elements. For example, the geometric relationships between the solid representations of BIM objects can be used as a BIM validation methodology in health and safety compliance checks [22]. Geometric relations among triangulated objects have also been used in automatic code compliance checking operations for fire safety [23]. The topological spatial relationships (adjacency, connectivity, and inclusion) among BIM objects have also been investigated using fast comparison rules on min/max point coordinates for code compliance checking in Zhao et al. [24]. As far as the implementation of code compliance rules is concerned, GRC is also utilized by various software tools like Solibri Model Checker [25] and specialized geometry engines such as FORNAX [26] and processing languages such as BERA [27]. GRC concepts have been utilized alongside Graph Neural Networks (GNN) to carry out BIM quality control tasks [28]. Additionally, GRC concepts have been applied for geometric quality checks (GQC) and the creation of GQC knowledge graphs to ensure code compliance in construction projects [29].

2.5. Knowledge graphs

The formation of knowledge graphs within the BIM context to perform data queries has received considerable attention [12,30]. Although the geometric context of BIM data can be described in a knowledge graph form [31], queries in this context are quite challenging. To address this, an ontology named "File Ontology for Geometric Formats", or FOG [32], has been created to connect files containing BIM geometric data, like glTF and OBJ files, with non-geometric knowledge graphs. Also, ways to interlink these different geometric files have been investigated with GEOM ontology [33]. Several ontologies have been developed to describe the geometric representation of solid objects including the ontology of geometric primitives [34], and the Ontology Managing Geometry [35]. Although the previous attempts at developing geometry-related ontologies, focus on the different geometric representations and their links, to this day, no ontology supports the GRC concepts.

As it is reported also in Wang et al. [36], Ouyang et al. [37], GRC concepts can be used to infer missing semantic links among BIM objects belonging to different data domains and enrich respective knowledge graphs with these links. These research efforts highlight the need to define geometric relations among BIM elements more formally according to a domain-agnostic format and to detect and generate instances of these relations by applying GRC operations to BIM element pairs. For example, the detected geometric relationships among geometric representations of HVAC element pairs can be used to form knowledge graphs [38], using the brick ontology [39]. These graphs capture the connectivity patterns of a building's HVAC network. The work presented here attempts to support such application of GRC concepts towards an HVAC topology generation in a knowledge graph format.

Furthermore, triggered by the potential applications of the GRC concepts, the need to store these relations in graph form, and the fact that there is no GRC ontology, one of the objectives of this work is to introduce such an ontology (see Section 3.5).

3. Methodology

Certain geometric relations among BIM elements' solid geometric representations can be used to detect missing semantic relationships. These geometric relations may include adjacency, clash, and containment relations. This section introduces a process for identifying these geometric relations using a dedicated tool called Geometric Relation Checker (GRC tool).

3.1. Process outline

To detect missing semantic relations among BIM elements using geometric means a four-step process is introduced—this process is outlined in the block diagram of Fig. 1. Input to the proposed methodology is a BIM model that includes geometric information, as would be exported from any authoring tool using the Design Transfer View Model-View Definition. To make the discussion more concrete, we consider an IFC model without loss of generality as input. The output of the process is the topological information, which can be exported in several formats. Such information can enrich the original IFC model semantically, a case that has been studied extensively. Rather, we focus on defining a compact representation using RDF and a lightweight ontology to define the detected geometric relationships. This compact representation can support different use cases including the one seen in the case study below. It should be mentioned that the serialization approach is less critical in extracting the topology itself, but the lightweight ontology presented in Section 3.5 can be useful in practical implementations.

1. Initially, the geometric content of the IFC file is extracted using the Geometry Exporter.
2. The extracted geometric content (from step 1), together with a list of GUIDs of elements to be checked and the check types, are fed to the GRC importer, which produces the input XML file of the Geometric Relation Checking tool (GRC tool).
3. The input data is processed by the Geometric Relation Checking tool. The checks are performed and the results are reported in an output XML file.
4. Using the output of step 3, the Knowledge Graph Generator can generate a final knowledge graph.

The output of the final step (step 4) of the previous process is a knowledge graph. This is achieved using the Knowledge Graph Generator component (see Fig. 1), which functions as an ETL process that transforms the data from the output XML file (step 3) into a knowledge graph, following a specific ontology, such as the one introduced in Section 3.5. Using additional reasoning, the Knowledge Graph Generator component can also generate a knowledge graph based on other ontologies, as well.

This knowledge graph contains the geometric relationships among all the checked pairs of elements. These relationships can be exploited in several ways. In this study, the detected geometric relations among the solid representations of building HVAC elements and building spaces will be used to infer the connectivity of these components, establish paths from source to sink elements and pinpoint the locations of HVAC terminal units within the building's spatial topology—crucial information for building energy performance simulations.

3.2. Geometric relation checking tool

The GRC tool which supports the checking of geometric relations is described in more detail in this section.

3.2.1. Description of operation

GRC is a software tool that checks the presence of certain geometric relations in pairs of input geometric representations of BIM elements defined broadly as definition shapes based on the IfcProductDefinitionShape specification. At the moment, three geometric relation checks are implemented and used in GRC: adjacency, clash, and containment. These relations are illustrated in Fig. 2, using the cubic solid geometric representations of an element pair (A, B).

The above geometric relations are defined as follows:

Adjacency: An adjacency is detected between two elements if their solid geometric representations share one or more common boundary surfaces. Such a common boundary surface, shared by the two solid geometric representations of an element pair (A, B), is displayed in blue in part A of Fig. 2. Appropriate threshold values are defined to detect adjacency, based on plane distance, as described in Section 3.4.

Clash: A clash between two elements occurs if their solid geometric representations intersect without one fully containing the other. An example of such a clash, involving the intersecting geometric representations of element pair (A, B), is shown in part B of Fig. 2. In this example, the clash surfaces are highlighted in blue.

Containment: A containment is detected between two elements if the solid geometric representation of one element is contained entirely inside the solid geometric representation of the other. A containment example, between the solid geometric representations of the elements (A, B), is demonstrated in part C of Fig. 2.

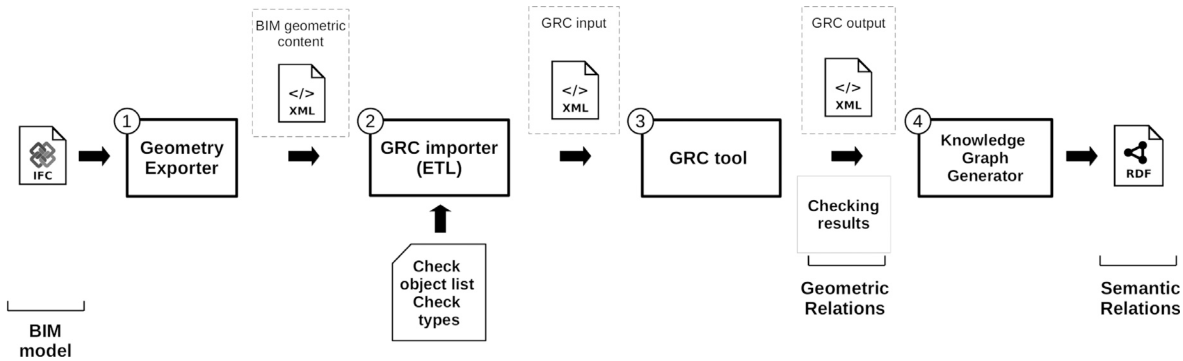

Fig. 1. Block diagram of the knowledge graph generation produced by BIM geometric relation checking process.

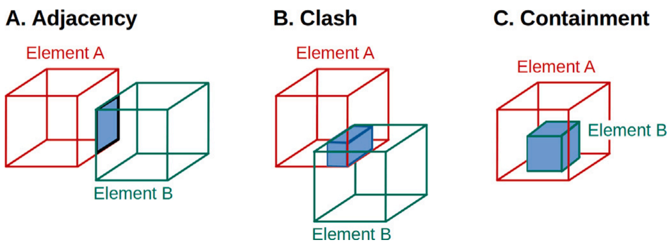

Fig. 2. Types of geometric relations between the solid geometric representations of two elements: (A) Adjacency, (B) Clash and (C) Containment.

3.2.2. GRC tool I/O specification

In general, the GRC tool performs pairwise geometric checks (as illustrated in Fig. 2), between elements belonging to either one element set $A = \{A1, A2, \ldots\}$ or two element sets $A = \{A1, A2, \ldots\}$ and $B = \{B1, B2, \ldots\}$. As illustrated in Fig. 3, in both input and output data structures of GRC, a finite number of checks are defined along with units of length and area measurement. The checks in the input are mapped to the checks in the output in a one-to-one fashion: the first input check (that contains the check types, check relations and the sets A, or set A of elements to be checked), is mapped to the first check of the output (that contains the results of the check operations: the detected check relations and their respective check relation surfaces), and so on. Both input and output data files of GRC tool are illustrated for a clash detection check, in parts A and B of Fig. 3, respectively. GRC has two additional output file exporting options: OBJ files for viewing purposes and BCF files to facilitate BIM collaboration. Next, we present GRC's input and output data structures.

GRC input. As mentioned above, the GRC input (illustrated in part A of Fig. 3.) contains multiple check classes, each comprising subclasses that define the geometric representations of the objects to be examined. These objects are grouped into element sets (A and B) and are associated with specific types of checks to be performed. These subclasses are:

CheckType: Defines the type of the check that is going to be performed. It can attain one out of two possible string values: “CrossCheck” or “SelfCheck”. If the CheckType has a “CrossCheck” value cross-checks among all elements of two sets A and B are going to be performed. If the CheckType has a “SelfCheck” value, checks among all possible pairs of elements belonging to a single set A, are going to be performed. In the last case, all commutative pairs will be checked once (checks (A_i, A_j) and (A_j, A_i) are the same, where $A_i, A_j \in A$).

CheckRelations: Contains one or more CheckRelation subclasses. Each CheckRelation contains the geometric relationship type to be checked, as a string attribute and additional parameters (if any) as additional attributes. Currently, three CheckRelation types are supported that are defined as the following string attributes: “Adjacency”, “Clash” and “Containment”.

ElementSetA: Contains information about the objects that belong to set A.

ElementSetB: Contains information about the objects that belong to set B. It exists only if the CheckType has the “CrossCheck” value.

The elements listed in the element sets A and B represent structured data points that refer to the properties of an object in a Building Information Modeling (BIM) environment, specifically using IFC (Industry Foundation Classes) standards. Each element contains data relevant to the 3D spatial configuration and geometric description of the respective BIM object, contained in the following data classes:

GID: Contains a unique, 25-character identifier called an IFC GUID (Globally Unique Identifier) as a string. It ensures each object being checked has a distinct ID, which is essential for referencing specific objects in the IFC model.

GlbLocations: Includes the location vectors of the object's placement transformations in 3D, listed as successive IfcLocalPlacement transformations. It is formatted as a string in the form:

“$[[l_{1x}, l_{1y}, l_{1z}], [l_{2x}, l_{2y}, l_{2z}], \ldots]$”, where each “$[lx, ly, lz]$” defines the coordinates of each transformation's origin point, which affects the object's position in 3D space.

GlbAxes: Stores the axis vectors for each placement transformation, also following successive IfcLocalPlacement transformations in 3D. The vectors are given as: “$[[a_{1x}, a_{1y}, a_{1z}], [a_{2x}, a_{2y}, a_{2z}], \ldots]$”, representing the direction of each axis. Each vector has a Euclidean norm of 1, meaning it is a unit vector, ensuring standard length and orientation consistency across transformations.

GlbDirections: Defines the direction vectors of each transformation in 3D, formatted similarly to axis vectors, with each direction vector “$[dx, dy, dz]$” also having a unit length (norm of 1). This standardization guarantees consistent orientation and rotation for each transformation in the object's local coordinate system.

DefinitionShape: One or more DefinitionShape classes describe the object's geometric form, including parametric (dimension-based) and non-parametric shapes. Each shape follows the IfcProductDefinitionShape format from IFC4 standards, providing a comprehensive geometric description for quality control. GRC (Geometric Relation Checking) tools support various DefinitionShapes, which represent different geometric representations needed for diverse BIM applications.

Together, these data classes form a structured way to represent and validate an object's 3D placement, orientation, and shape in IFC-based BIM models, which is essential for automated checking and quality control.

GRC output. As mentioned above, the GRC output (illustrated in part B of Fig. 3.) contains multiple check classes, each comprising of subclasses that contains the results of the checks defined in the input. These subclasses are:

CheckRelation: This class can be either: Clash, Containment, or Adjacency, depending on the detected geometric relation for the respective check. In the example of part B of Fig. 3 a Clash is presented as a detected CheckRelation element. Each CheckRelation element contains the following string attributes:

- The check ID number as Check_ID,
- The internal element ID number of the first element involved in the detected CheckRelation from the set A, as ElementA_ID1.
- The IFC GUID of the first element involved in the detected Check relation from set A, as ElementA_GID1.
- The internal element id number of the second element involved in the detected CheckRelation, which is defined either: in set A as ElementA_ID2 if the CheckType of the Check is defined as a "SelfCheck" in the input, or in set B as ElementB_ID2 if the CheckType of the Check is defined as a "CrossCheck" in the input.
- The IFC GUID of the second element, involved in the detected CheckRelation, which is taken either from set A as ElementA_GID2 if the CheckType of the Check is defined as a "SelfCheck" in the input, or set B as ElementB_GID2 if the CheckType of the Check is defined as a "CrossCheck" in the input.

Each CheckRelation subclass contains a CheckRelationSurfaces subclass that contains the surfaces if any, of the CheckRelation, as described next.

CheckRelationSurfaces: Depending on the type of the detected CheckedRelation, the CheckRelationSurfaces can be one of the following: ClashSurfaces, AdjacencySurfaces, or ContainmentSurfaces. In the example presented in part B of Fig. 3, since the detected CheckRelation is a Clash the CheckRelation surfaces are defined as ClashSurfaces. Each of the detected CheckRelation surfaces are planar polygons, they are defined using one outerContour element, describing the outer perimeter of the respective polygon in 3D, and possible multiple innerContour elements describing its holes in 3D, if any. Each of the outer- and innercontour elements contains a set of points in 3D as elements point3D. Finally, each of the point3D elements contains the coordinates of the respective point in 3D as x,y, and z string attributes.

To give an example of the previous I/O specification, if the check type is a "CrossCheck", the check relation is "Clash" $A = \{A1, A2, A3\}$ and $B = \{B1, B2\}$ the following $3 \times 2 = 6$ pairs of elements will be checked for potential clashes:

$\{(A1, B1), (A1, B2), (A2, B1), (A2, B2), (A3, B1), (A3, B2)\}$.

Alternatively, if in the previous input configuration, the check type is "SelfCheck" without the element set B, then the pairs that will be checked for potential clashes will be:

$\{(A1, A2), (A1, A3), (A2, A3)\}$.

3.3. GRC implementation and performance

At the current stage, GRC receives as input the solid geometric representations of BIM elements that are defined according to the IfcProductDefinitionShape class. The GRC process consists of three primary steps, executed sequentially by different subroutines. These steps include two input/output interface steps (1 and 3) and a single core step (2):

1. Boundary Surface Extraction: In the first step, boundary representations of the elements to be checked are extracted from the parsed input XML file. This involves converting the parametric and non-parametric geometric descriptions of the elements' solids (under the DefinitionShape class) into sets of boundary surface polygons (B-reps). These polygons follow the outward normal convention, where the normal vectors of all boundary surface polygons point outward from the solid. At the end of this step, the element checking pairs, along with the defined checking operation type in the input, and the generated boundary surface sets, form separate computation threads to be executed in the next core step.
2. Main checking operations: The computational threads formed in step 1, which contain the elements to be checked, their corresponding boundary surface sets, and the checking operation type, are executed in parallel. The algorithmic processes of these checking operations are described analytically in Lilis et al. [40].
3. Output: The results of the checking operations performed in parallel during step 2 are compiled into an output, which can be either a native XML file, an OBJ file for visual inspection, or a BCF file for BIM collaboration.

GRC aims to be an extremely fast checker, which is why it is implemented in C++ and utilizes multithreading by assigning different element pairs to separate processor cores for checking. To accelerate the processing operations further, GRC handles the clash-dependent and time-consuming operations (such as clash and containment) differently from the less demanding operations (like adjacency).

For the clash-dependent operations, GRC does not rely on proximity criteria (like distance) to eliminate certain pairs from consideration. Instead, it only checks element pairs whose Axis Aligned Bounding Boxes (AABBs) intersect. AABB clash detection involves minimal multiplications and condition checks compared to standard clash detection, making it significantly faster. When AABBs intersect, regular clash detection is then performed using Binary Space Partitioning (BSP) trees (see [41]). To further optimize computation time, GRC calculates and preloads the BSP trees and AABBs of all elements during the Boundary Surface Extraction (Step 1) into memory for use in the clash-dependent operations.

The execution time of GRC depends on the number of element pairs to be checked and the resolution of the elements (i.e., the average number of boundary surfaces per element). As demonstrated in the following application section, with this implementation, GRC can conduct clash-dependent operations on millions of element pairs of relatively moderate geometric resolution, within a matter of minutes.

3.4. GRC accuracy

The accuracy of the GRC is influenced by the complexity of the solids that are being checked. As outlined in Lilis et al. [40], the clash and containment checks performed by GRC rely on set operations (intersection) on polyhedra using Binary Space Partitioning (BSP) trees, that are created through polygon clipping operations in three dimensions. These clipping operations (implemented using [42]) involve pairs of polygons in 3D space, where one acts as the clipping polygon and the other as the clipped polygon. The accuracy of these operations is determined by the angle between the normal vectors of these polygons. The closer this angle is to zero degrees, the less accurate this clipping operation is. Consequently, the accuracy of GRC operations decreases when neighboring polygons in the boundary representations become nearly coplanar. Such conditions often occur in polyhedral solid boundary representations (b-reps) with low gradient values at their edge points (i.e., b-reps with smooth boundary surfaces). Therefore, the GRC's clash and containment detection is expected to be less accurate for solids with smooth boundary surfaces compared to those with sharper edges.

For performing geometric operations at different scales, particularly for adjacency checks, it is essential to establish a coplanarity condition for polygon pairs. This condition should be defined by an angle and distance threshold across all GRC operations. This coplanarity condition and the related distance and angle thresholds is a prerequisite for the GRC adjacency checks as described next. Adjacency checks involve

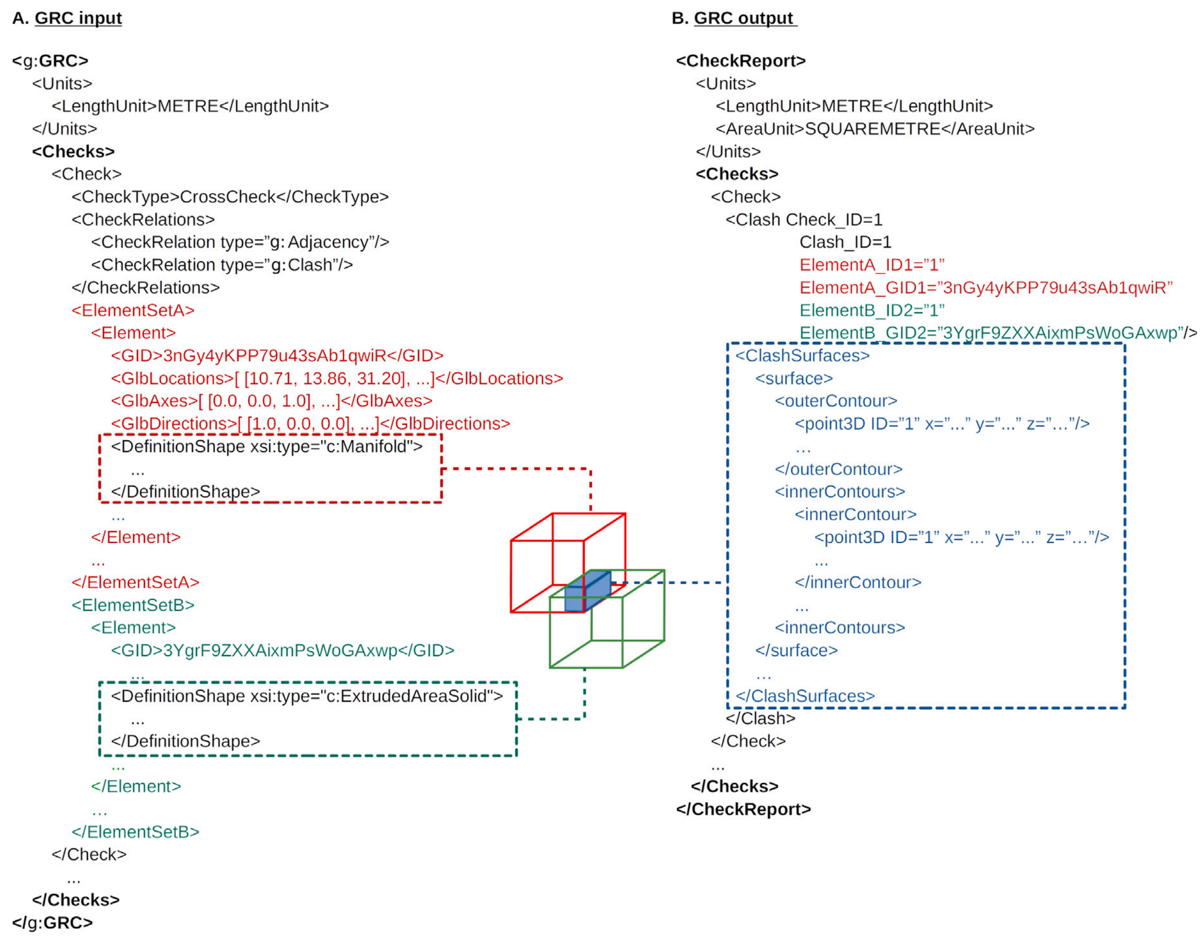

Fig. 3. Input and output XML file snippets of GRC tool.

pairwise polygon checks, as illustrated for polygons A and B in Fig. 4. Specifically, each boundary representation polygon of the first solid is checked with a corresponding boundary representation polygon of the second solid, as outlined by the function F_{cos} found in Lilis et al. [40]. The angle threshold for these checks is defined as the allowable deviation from 180 degrees between the normal vectors of the two checked polygons. If this deviation is within this threshold, the polygons are deemed to be parallel with opposite orientations. This angle threshold is Φ^{th}, as illustrated in Fig. 4. The detected parallel polygons with opposite orientations form candidate polygon pairs. For each candidate pair, the maximum distance between a point on one polygon and the plane of the other (plane-to-point distance) is checked against the distance threshold. Using the previously defined angle and distance thresholds and according to Fig. 4, adjacency is detected if three conditions are met: (a) the polygons are parallel ($\phi < \Phi^{th}$) with opposite orientations, (b) the maximum point-to-plane distance remains below the distance threshold ($d_{max} < D^{th}$), and (c) the projections of the polygons intersect on a plane parallel to either one of the polygons' planes (see Fig. 4). When these criteria are satisfied, the polygons and the respective solids are considered adjacent. In case only a and b are satisfied the polygons are considered coplanar without being adjacent.

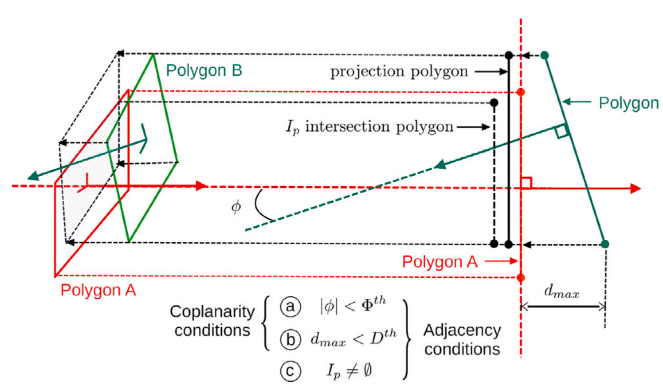

Fig. 4. Illustration of angle and distance tolerance thresholds, used in adjacency relations.

3.5. GRC ontology

To store the obtained GRC results in a knowledge graph form (step 4 of the methodology presented in Fig. 1), we introduce an initial draft of a grc ontology that uses a single `grc:SolidModel` class as illustrated with a blue box in the block diagram of Fig. 5. As displayed in this figure, the grc ontology has bindings with classes from the brick [39] (`brick:Equipment`) and FSO ontologies [43] (`fso:Component` and `fso:Segment`), via the `grc:hasSolidModel` relationship. Additional links of the `grc:SolidModel` class with the `geom:Geometry` class of the Ontology of Geometric Primitives [34] via the `grc:hasGeometry` relation, can be established. Depending on the detected geometric relationship, each `grc:SolidModel` instance is connected with another `grc:SolidModel` instance. If adjacency or clash is detected, the two `grc:SolidModel` instances are connected with the `grc:isAdjacentTo` or `grc:intersectsWith` relationship, respectively. If containment is detected, the `grc:SolidModel` instances are connected with the `grc:isContainedIn` or `grc:contains` relationships.

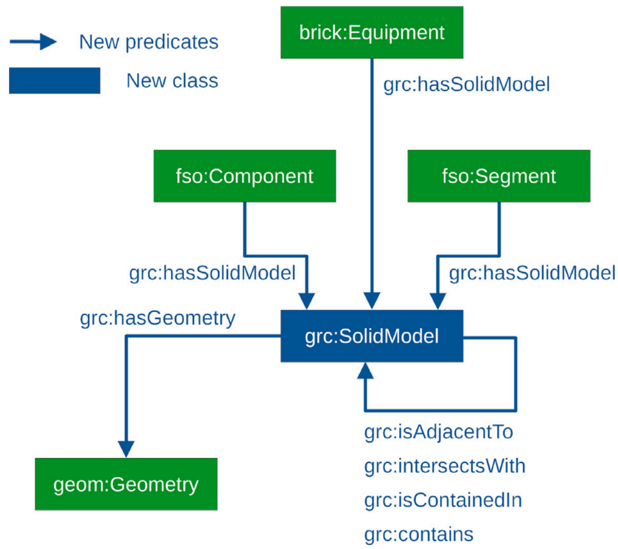

Fig. 5. UML diagram of GRC ontology and its links to REC and FSO ontologies.

4. Application in MEP fluid transfer networks

In this section, we showcase how geometric relationships among the solid geometric representations of BIM elements can lead to the discovery of missing BIM element relations and their corresponding knowledge graph links (logical relations). This approach is applied explicitly to data within the building MEP domain because the BIM elements have complex geometric representations. The approach can be easily extended to other domains, such as architecture, where the geometrical representations are simpler. While geometric relations can be utilized to unveil desired BIM element relations [38], it is important to note that not all existing BIM element relations can be identified using geometric methods due to various reasons [44,45]. Recent research efforts have been increasingly directed towards generating MEP topology in knowledge graph format from BIM data, leveraging BIM geometric representations [38,46], as well as employing rule checking and subgraph matching techniques [47]. Moreover, the geometric relations identified through this process can be employed to uncover design errors, such as clashes among the geometric representations of MEP elements.

Within BIM models, various interconnected MEP elements form distinct fluid transfer networks. The fluid being transferred could be categorized as air or gas, water, waste, or electricity. These elements serve three primary roles within their networks: they are either source elements, distribution elements, or terminal elements. Source elements produce fluid resources, distribution elements transfer them, and terminal elements consume them. The proposed workflow aims to facilitate the generation of knowledge graphs that depict the connections of MEP fluid transfer networks. This is achieved by identifying missing semantics in BIM data through the geometric relationships among the solid representations of BIM MEP elements and building space volumes. Before delving into how these missing semantics can be inferred, the complete MEP fluid transfer network and its BIM data requirements are defined in the subsequent subsections.

4.1. Complete MEP fluid transfer network

From a data completeness perspective, three conditions should be satisfied for an MEP transfer fluid network to be complete:

1. All MEP terminal elements should be fed into a building space.
2. Source elements should be connected with terminal elements via a series of distribution elements.
3. A flow relation should be defined for every network edge connection between source and terminal elements.

4.2. BIM data requirements

To generate the previously defined complete MEP fluid transfer networks using data from BIM data sources, the available BIM data should satisfy the following conditions:

1. All MEP elements should have solid geometric representations.
2. The solid geometric representations of the connected MEP elements should share common boundary surfaces or intersect with each other.
3. At least one element at every edge of the MEP knowledge graph should have a predefined flow direction.
4. The geometric representations of MEP terminal units should intersect with at least one building space volume.

4.3. Primary and derived knowledge gaps

To construct a comprehensive MEP fluid transfer network, it is essential to explore the available information within a BIM model. This information can be categorized into two distinct "knowledge sets": the primary knowledge set and the derived knowledge set. The primary knowledge set comprises information specified by the BIM designer and exported within the BIM model. This data is indispensable for creating a complete MEP fluid transfer network and cannot be deduced from other related data. Conversely, the derived knowledge set encompasses information that can be inferred from other data contained within the primary knowledge set. Any gaps in the primary knowledge dataset may lead to violations of the conditions outlined in the preceding subsection.

Examples of gaps in both primary and derived knowledge data are depicted in Fig. 6 for a serial connection of three pipe segments: A, B, and C. In part A of Fig. 6, an entire pipe segment (Segment B) is absent from the BIM data. This segment serves to connect pipe segments A and C, making the knowledge of segment B crucial for generating a comprehensive knowledge graph. Thus, the absence of segment B serves as an instance of a primary knowledge gap, violating condition 2 of the data requirements outlined in the previous subsection.

In the example shown in part B of Fig. 6, although the geometric representation of segment B exists within the BIM model, the semantic connections between segments A and B, and between B and C, are not explicitly contained in the BIM data. In this scenario, these absent semantic connections can be inferred using geometric operations on the solid geometric representations of the involved elements. These missing connections represent an instance of a derived knowledge gap since they can be uncovered using geometric methods. By identifying and generating such missing semantic links (illustrated in red in Fig. 6), the completeness of the MEP knowledge graph and the respective MEP fluid transfer network can be restored.

After presenting the BIM data requirements two BIM relation types will be discussed in the following subsection. These relation types will play a fundamental role in the formation of complete MEP transfer fluid networks.

4.4. MEP relation types

This work considers two distinct types of BIM relations involving MEP elements and their corresponding building space volumes: connectivity relations among the MEP elements themselves and containment relations between the MEP elements and the volumes of their building spaces. These two types of BIM relations and their associations with geometric and semantic relations are elaborated upon in the subsequent subsections.

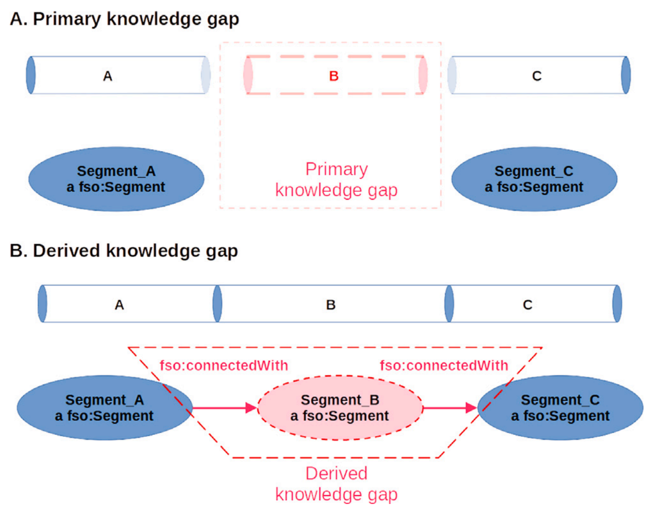

Fig. 6. Primary and derived knowledge gap examples.

4.4.1. MEP connectivity

Each element within an MEP network, whether it is a source, sink, or distribution element, is linked to another element within the same network. This connection manifests in three distinct contexts: within a BIM framework, a geometric framework, and a knowledge graph framework. The connectivity of MEP elements is represented in BIM files through specific class relationships between pairs of connected elements, as depicted in part A of Fig. 7. The BIM connectivity relation, which represents a topological relation, is implemented through the IfcRelConnectsPorts relation within the openBIM IFC schema.

In a geometric context, the solid geometric representations of connected MEP elements either share common boundary surfaces or slightly intersect, meaning their solid volumes have a relatively small common part. In this context, MEP connectivity is established either as a geometric adjacency relation (as depicted in part B of Fig. 7) or a geometric clash relation. The adjacency relation is exemplified in part B of Fig. 7 by the grey circular surfaces, which represent the common boundaries between the geometric representations of the pipe, the boiler, and the sanitary terminal units. Lastly, the MEP connectivity relation can be articulated as a topological relation within a knowledge graph context. This is demonstrated in part C of Fig. 7, where knowledge graph links are established using the `fso:connectedWith` relation (defined in the `fso` ontology [43]), linking a `brick:boiler` class element and a `brick:Equipment` class element via a `fso:Segment` class element (defined in the `fso` ontology [43]). In case the proposed grc ontology is used the predicates `grc:intersectsWith` and `grc:isAdjacentTo` should be mapped to the predicate `fso:connectedWith` when both subject object refer to a MEP distribution element.

Although the MEP connectivity relations, in knowledge graph terms, can be induced in a straightforward manner from the connectivity relations in a BIM context, this is not valid for the connectivity relations in a geometric context. For example, if two elements are geometrically adjacent to each other (their solid geometric representations share a common boundary surface) or intersect with each other (their solid geometric representations share common solid volumes), are not necessarily connected in a BIM, or a knowledge graph context, and vice versa. This inconsistency is due to errors in the involved manual design and BIM exportation processes, that produce the geometrical representations of the MEP elements. Table 1 presents different cases of geometric and BIM connectivity relation scenarios. According to Table 1, if two elements are geometrically connected (their solid representations are adjacent or intersect) and are not connected in BIM, there is either a BIM design/modeling error or the two elements are adjacent

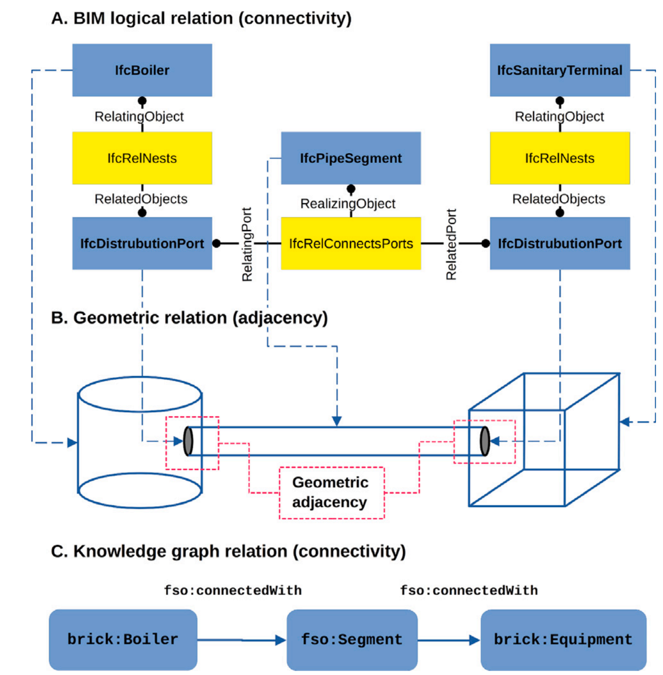

Fig. 7. Connectivity expressed using three relation types: (A) BIM, (B) Geometric, and (C) Knowledge graph.

Table 1
Confusion matrix related to connectivity relation detection in geometric and BIM contexts.

BIM Connectivity / Geometric Adjacency/Clash	Yes	No
Yes	(Issue free)	Adjacent Elements OR Design/Modeling error (false positive)
No	Design/Modeling or exporter error (false negative)	(Issue free)

to each other without being connected (false positive case). Finally, if two elements are connected in the BIM and not geometrically connected (there is no adjacency or clash), there is a BIM design/modeling error or a BIM exporter error, causing the geometrical representations of the elements to clash with each other or to be far apart (false negative case).

4.4.2. MEP-space containment

The different MEP elements found within the BIM data may be associated with a particular building space. This association is depicted in BIM models through specific class relationships. For example, in IFC models, as demonstrated in part A of Fig. 8, a containment relationship is defined between the classes IfcBoiler and IfcSpace, indicated by the IfcRelContainedInSpatialStructure connection. This relationship can also be represented in knowledge graph terms using the predicate `brick:isLocationOf` from the `brick` ontology as illustrated by the triplet displayed in part C of Fig. 8. In this example, the predicate links the subject `brick:Space` (which is assigned to the IfcSpace class) to the object `brick:Boiler` (which is assigned to the IfcBoiler class). Finally, these connections can be identified through geometric clash and containment detection procedures applied to the solid geometric representations of the elements involved (such as the volumes of the building space and its contained element). As illustrated in the example provided in part B of Fig. 8, the solid geometric representations of the boiler and the space volumes intersect, indicating a geometric

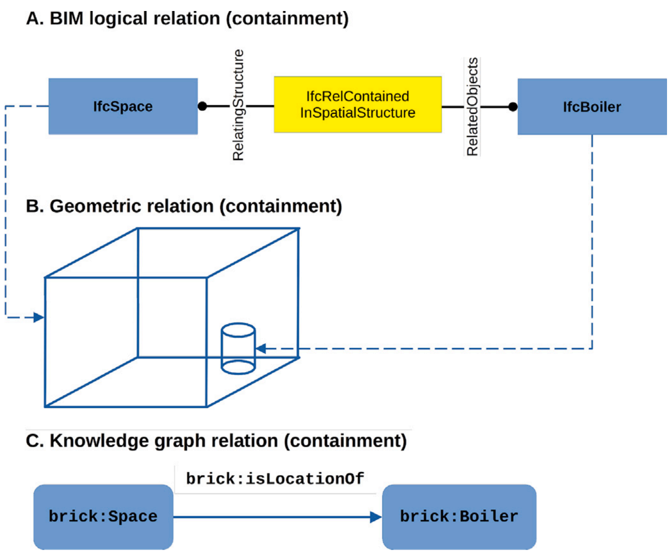

Fig. 8. Containment expressed using three relation types: (A) BIM, (B) Geometric, and (C) Knowledge graph.

Table 2
Confusion matrix related to containment relation detection in geometric and BIM contexts.

Geometric Containment/Clash \ BIM Containment	Yes	No
Yes	(Issue free)	Design/Modeling error (false positive)
No	Design/Modeling or exporter error (false negative)	(Issue free)

containment relation (where the boiler solid is entirely within the volume of the space solid). If the MEP element does not fit fully inside the space volume, the intended containment relation can still be identified using geometric clash detection instead of geometric containment detection. For this reason, if the proposed grc ontology is used, the predicates `grc:isContainedIn` and `grc:intersectsWith` should be assigned to the predicate `brick:isLocationOf` if the subject refers to a building space and the object to a MEP terminal unit or system.

As in the case of the adjacency relations detected in BIM, for the containment relations, there are four possible cases where the geometric clash or containment relation and the containment in BIM might or might not be present. These cases are illustrated in Table 2. According to Table 2, if an MEP terminal element is contained in a space volume in a geometric sense (their respective geometric solid representations intersect, forming a clash or a containment, where one volume is contained inside the other) and there are no respective containment relations in the BIM model, then there is a BIM design/modeling error related to these elements (false positive case). Finally, if an MEP terminal unit is connected, via a containment relation in the BIM model, with one building space, but there is no containment or clash relation in the geometric sense (their geometric representations do not intersect), there is a design/modeling error or a BIM exporter error, causing the geometrical representations of the MEP and space elements to be far apart (false negative case).

4.5. Correlation between BIM relations and geometric relations

Based on the discussion of the previous sections, a BIM relation may involve one or more geometric relations. More specifically, one or more

Table 3
Connections between BIM semantic relations and Geometric relations.

Geometric relation \ BIM relation	Connectivity	Containment
Adjacency	✓	
Clash	✓	✓
Containment		✓

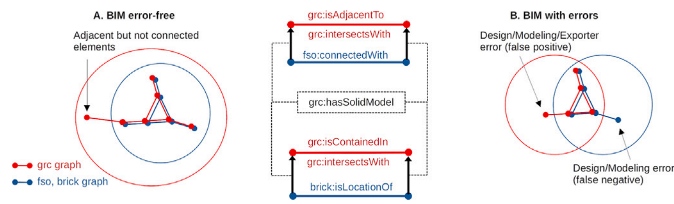

Fig. 9. Correlation between design/modeling errors and knowledge graph misalignment.

of the above geometric relations (adjacency, clash, and containment) can be used to reveal BIM semantic relations (connectivity and containment) among BIM elements. Table 3 illustrates the correlation between BIM connectivity and containment semantic relations and respective geometric relations.

In conclusion, as shown in part A of Fig. 9, a well-designed MEP BIM produces a semantic connectivity and containment knowledge graph—based on the fso and brick ontologies according to part C in Figs. 7 and 8 - that aligns with a corresponding geometric relation knowledge graph based on the grc ontology, except from the geometric adjacency links that correspond to elements that are adjacent with respect to geometry but not connected in semantic terms. Specifically, each edge in the semantic graph (displayed in blue in Fig. 9 indicating connectivity or containment) matches an edge in the grc-based graph (displayed in red in Fig. 9indicating adjacency, clash, or containment) and connects the same elements, apart from the aforementioned geometric adjacent relations. The red dots in Fig. 9 indicate `grc:SolidModel` classes and the blue dots in the same figure indicate brick or fso classes related to physical objects (`brick:Equipment`, `fso:Segment`, ...). In graph theory terms this graph alignment can be expressed as: $\mathcal{G}_s = \{V_s, E_s\} \subseteq \{V_g, E_g\} = \mathcal{G}_g$, where V is the set of graph nodes, E is the set of graph edges, $\mathcal{G}_g$ is the grc graph and $\mathcal{G}_s$ is the semantic (fso and brick based) graph. In contrast, as illustrated in part B of Fig. 9, mismatches between these graphs can typically be attributed to BIM design/modeling errors (false positives, $E_g \backslash E_s$) or errors in BIM design, modeling or export processes (false negatives, $E_s \backslash E_g$).

4.6. Geometry-induced knowledge graph establishment

This section investigates how previous geometric relations (adjacency, clash, and containment), which are correlated to MEP connectivity and containment relations, can be used for the formation of a geometry-induced knowledge graph among the elements in MEP-BIM models. This graph formation involves two stages, the GRC-based workflow stage and an ontology stage.

4.6.1. GRC-based workflow for BIM relation inference

A GRC-based workflow was proposed to identify the existing connections among MEP entities in the MEP-IFC files. It has three stages to extract the connectivity pairs by checking the BIM context, executing the GRC tool, and implementing axis-aligned bounding boxes (AABB) checking operation. Fig. 10 illustrates the key steps in each stage.

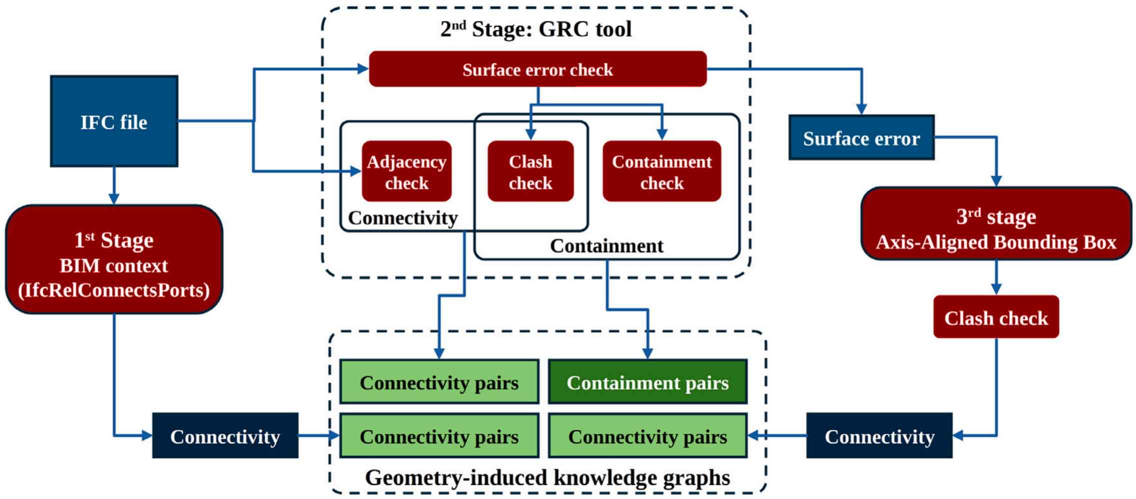

Fig. 10. Diagram of the proposed GRC-based workflow.

Connectivity relations. This workflow initially identifies all semantic connections by checking `IfcRelConnectsPorts` in the IFC file, generating a list of connectivity pairs. Subsequently, the GRC tool uncovers additional geometric relationships among the MEP elements and then exports the clash/adjacency pairs to create the connectivity pairs. Meanwhile, the GRC tool detects elements with geometric surface errors. To avoid incorrect GRC detections, the pairs of elements that have geometric errors, are forwarded to AABB checking to detect additional potential connectivity relations. Finally, all connectivity pairs are merged to facilitate the extraction of most linked pairs from IFC files, which provides the linking entities in the geometry-induced MEP knowledge graphs.

Containment relations. Containment relationships are common in BIM models, particularly between systems (e.g., AHUs, FCUs, MVHRs) and spaces, as well as between terminals (e.g., diffusers, grilles, radiators, door heaters) and spaces. However, these containment relations cannot be inferred solely through the BIM context or AABB checking. Only the GRC tool can reliably infer them by examining (2nd stage of Fig. 10) potential clash and containment relationships between the geometric representations of spaces and MEP systems and terminal units. The containment pairs generated at this stage help to identify terminal and system locations within the building's spatial layout. These pairs, along with connectivity pairs, are then used to create geometry-based MEP knowledge graphs.

4.6.2. Ontology mapping

Ontologies offer a comprehensive means to semantically describe all the elements and their relationship within the architectural and MEP domains. In this work, two ontologies were introduced and integrated to cover the MEP elements as much as possible. brickSchema [39], developed by Brick Consortium, effectively delineates building assets and their interconnections, especially within the architectural domain. As a supplement, Flow Systems Ontology (FSO) [43], concentrating on interconnected systems and energy flow connections, was introduced to represent the pipework and ductwork segments. Table 4 illustrates the mapping relationship between IFC classes and ontologies. Regarding the three geometric relationships between the MEP elements, `fso:connectedWith` indicates the connectivity relations (clash and adjacency), and `brick:isLocationOf` indicates the containment relations. To prevent naming conflicts, the IFC global ID (GUID) was used to represent each MEP element in the geometry-induced knowledge graphs.

Finally, it is important to note that when HVAC systems or terminal units are described by the generic `IfcBuildingElementProxy` class, they cannot be identified as HVAC terminal or system elements, which introduces additional challenges in generating the final MEP knowledge graph. However, in certain cases, specific properties of a system or terminal unit may be available, assisting in its identification.

Table 4
Mapping between IFC and Ontology classes.

IFC class	Ontology:Class
IfcSpace	brick:Space
IfcChiller	brick:Chiller
IfcHeatExchanger	brick:HX
IfcBoiler	brick:Boiler
IfcAirTerminal	brick:Air_Diffuser
IfcFan	brick:Fan
IfcDuctSegment	fso:Segment
IfcDuctFitting	fso:Fitting
IfcPipeSegment	fso:Segment
IfcPipeFitting	fso:Fitting
IfcFlowMeter	brick:Meter
IfcValve	brick:Valve
IfcWasteTerminal	brick:Equipment
IfcUnitaryEquipment (PredefinedType)	brick:AHU brick:DOAS brick:FCU
IfcAirTerminalBox	brick:VAV
IfcSpaceHeater	brick:Radiator
IfcRelConnectsPorts	fso:connectedWith
IfcRelContainedInSpatialStructure	brick:isLocationOf

5. Examples

5.1. Overview

We demonstrate the application of the introduced GRC tool based methodology, on the MEP system of the Podium of the One Pool Street (OPS) building on the University College London campus. The building incorporates a sophisticated Mechanical Ventilation System designed to meet specific requirements. This system ensures the circulation of fresh, tempered, or cooled air throughout the building's spaces while simultaneously removing exhaust air. These functions are facilitated by air handling units, which are integral to the building's infrastructure and are linked to both the Low-Temperature Hot Water (LTHW) heating system and the Chilled Water (CHW) system. The LTHW Heating System provides heat, while the Chilled Water System offers cooling, contributing to a comfortable indoor environment.

The air handling units feature heat/cool recovery mechanisms, which extract energy from exhaust air, subsequently reducing the

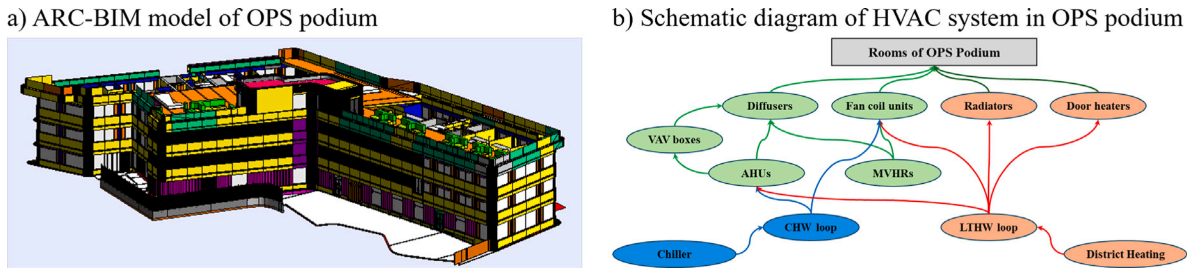

Fig. 11. (a) 3D visualization of OPS podium, and (b) schematic diagram of its HVAC system.

heating and cooling demands placed on the system via VAV boxes or constant air volume diffusers. Additionally, Mechanical Ventilation Heat Recovery (MVHR) units are strategically positioned to distribute air directly to various areas within the building. These units are interconnected with ceiling diffusers or fan coil units via a network of ductwork.

Furthermore, the building's Chilled Water Service (CHWS) is facilitated by an air-cooled chiller, which plays a crucial role in cooling the fresh air circulated by the air handling units. The CHWS system includes cooling coils within the air handling units and fan coil units dispersed throughout the building. These fan coil units not only recirculate the supply air but also further cool it before distributing it through secondary ductwork to ceiling diffusers. In terms of heating, the LTHW Heating System receives heat energy from a District Heating System and plate heat exchangers. This system provides warmth to the building's Podium via heater batteries located within the air handling units and fan coil units. Additionally, it serves to temper the supply of air provided by the Mechanical Ventilation Systems. Radiators, warm air curtains, over-door heaters, and heat interface units (HIUs) also leverage the LTHW Heating System to provide localized heating to specific areas within the building. Fig. 11 presents the architectural model of the building and its HVAC schematic diagram. Similarly, Fig. 12 displays the visualization of the selected and exported IFC Architectural and MEP content.

Complementing these systems is the Mains Cold Water Service (MCWS), which is distributed to the building's Sprinkler System tank and a main bulk storage tank. A Booster Cold Water Service (BCWS) is then delivered to various points within the building, including the Hot Water Service. Booster sets equipped with electromagnetic water conditioning devices ensure efficient water distribution throughout the building, with each floor served by a dedicated branch of the BCWS.

Lastly, the building's Above Ground Rainwater Drainage Systems consist of strategically installed rainwater outlets located on the roof and terrace areas. These outlets discharge rainwater into Below Ground Rainwater Drainage Systems, which are installed by the Main Contractor, completing the building's comprehensive water management infrastructure.

5.2. Data processing

An ideal scenario entails inputting one solitary MEP-IFC file into the GRC workflow to establish an HVAC knowledge graph. However, the nature of the input geometric data, characterized by LOD 300–400 granularity, poses a challenge due to memory issues when parsing and checking large IFC files. More specifically, the number of pair checks for N elements grows polynomially ($O(N^2)$) (for N elements, the number of pair checks is $\frac{N(N-1)}{2}$). In the application example, $N \approx 20.000$, and thus the number of pair checks approaches 200 million pairs. Consequently, dividing the large complete MEP-IFC file into three smaller IFC files becomes necessary: two IFC files for the HVAC domain and one IFC file for the non-HVAC domain. The HVAC-related IFC files are for the air-loop elements and the water-loop elements (heating and cooling purposes), respectively. The non-HVAC IFC file covers the pipework

Table 5
Error detection rules for MEP elements.

MEP element	Rule
IfcPipeSegment	<2 linked elements
IfcPipeFitting	<2 linked elements
IfcDuctSegment	<2 linked elements
IfcDuctFitting	<2 linked elements
IfcValve	<2 linked elements
IfcFlowMeter	<2 linked elements
IfcPipeSegment	linked to (≥ 1) IfcDuctSegment
IfcPipeSegment	linked to (≥ 1) IfcDuctFitting
IfcPipeFitting	linked to (≥ 1) IfcDuctSegment
IfcPipeFitting	linked to (≥ 1) IfcDuctFitting
IfcAirTerminal	linked to (0 or >2) IfcDuctSegment
IfcAirTerminal	linked to (0 or >2) IfcSpace
IfcSpaceHeater	linked to (0 or >2) IfcSpace

elements for domestic hot/cold water systems and rainwater/drainage systems. This splitting approach can reduce the inputted IFC files with minimum manual work. Fig. 12 illustrates the ARC-BIM and MEP-BIM models in Revit, and the exported IFC files are shown as well.

5.3. MEP knowledge graph establishment

This work uses a building-oriented ontology, named BrickSchema, to develop MEP knowledge graphs for achieving error detection, bridging from air terminals to systems, and splitting the MEP models into different domains. To accomplish a number of processing steps using the GRC process, are required, as outlined in the following subsections.

5.3.1. Error detection and path-finding

This GRC tool-based workflow can identify the majority of the connections among MEP elements, but some of them might be missing due to the data quality of the input MEP-IFC files. To ensure the completeness and accuracy of the generated MEP graph, it is essential to pinpoint the existing and potential issues/errors in the IFC file. For these reasons, a set of error detection rules for MEP elements are presented in the next Table 5 (links may represent a "Clash", an "Adjacency", or a "Containment" in the case of IfcSpace elements), is established.

When MEP graphs are generated, error detection can be performed to pinpoint the precise location and type of errors in the provided IFC files. In the OPS building podium case study, five error types were identified: (a) abnormal clash, (b) missing element, (c) geometric mismatch, (d) classification error, and (e) Space containment issue, as illustrated in Fig. 13. Many of these errors originate from inaccuracies in manual modeling during the design phase, bugs during the IFC exporting process or during the extraction of MEP models from the comprehensive federal BIM model of the entire building. Despite the identification and localization of errors, manual intervention is still required for resolution. One approach involves directly adding missing links or deleting incorrect ones in the MEP graph. Alternatively, the IFC file can be refined and corrected, and the updated version can be re-imported into the GRC-based workflow.

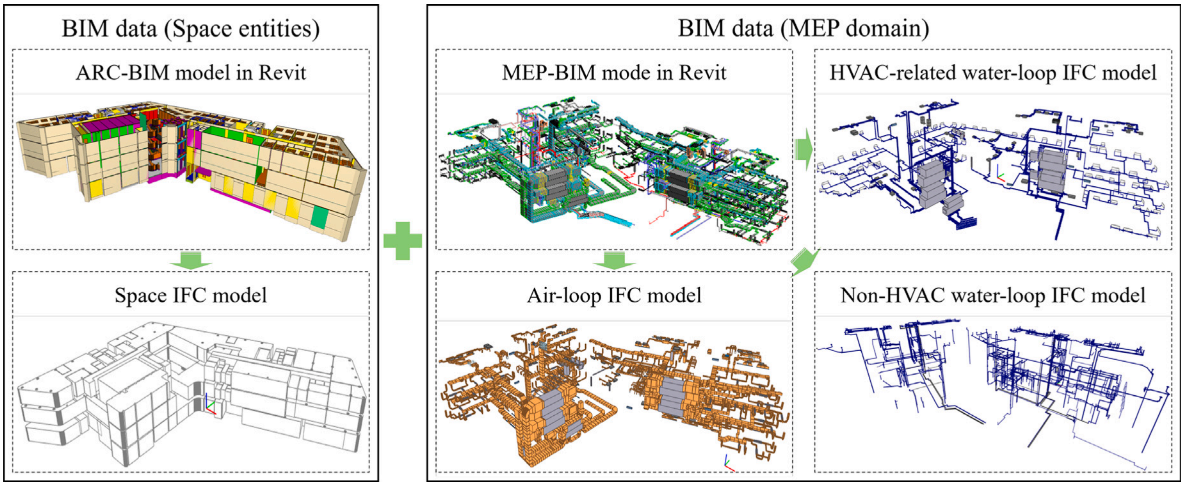

Fig. 12. Object selection in ARC/MEP Revit models and IFC exportation.

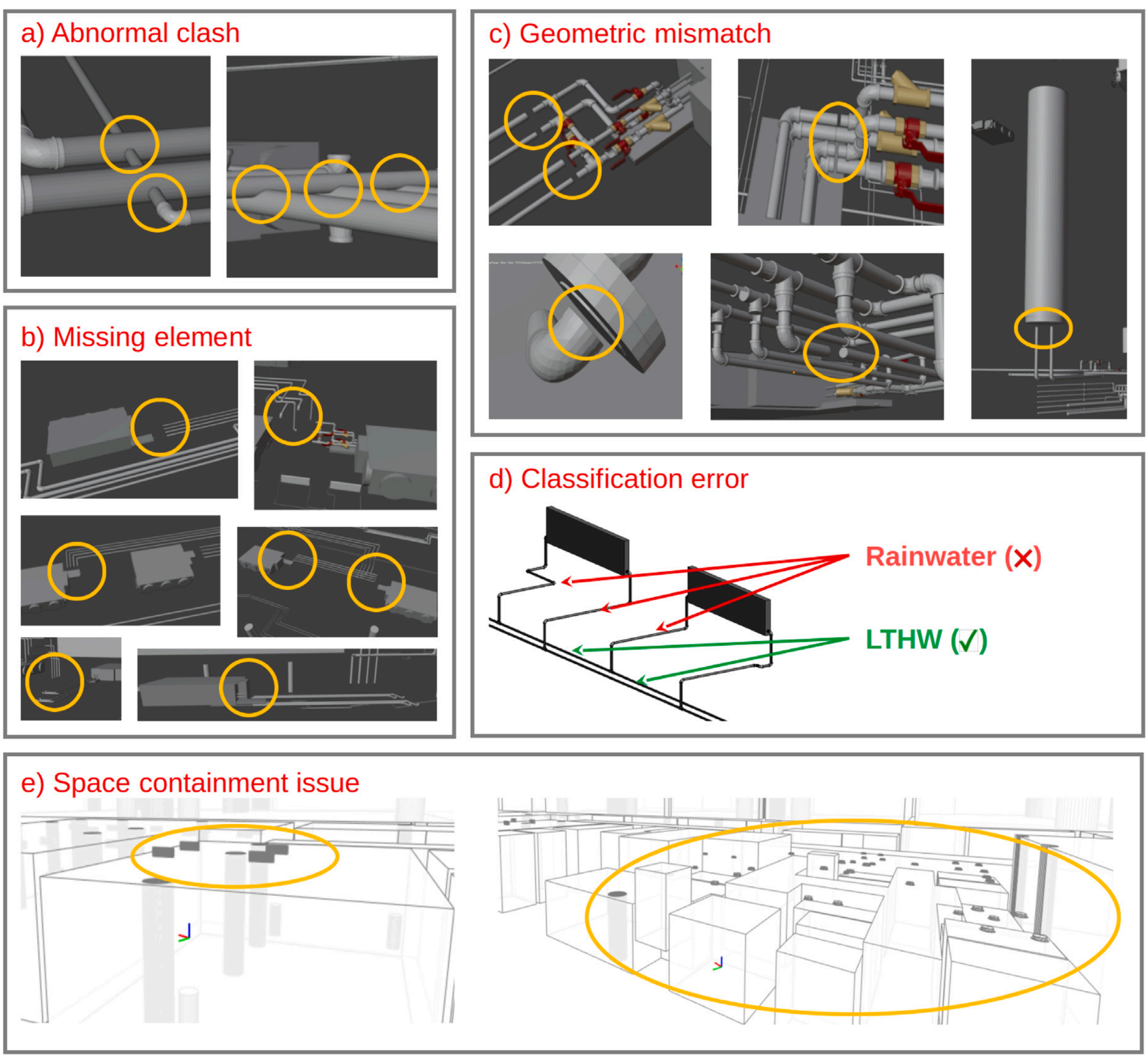

Fig. 13. Case study's BIM error classification: (a) abnormal clash, (b) missing element, (c) Geometric mismatch, (d) Classification error, and (e) Space containment issue.

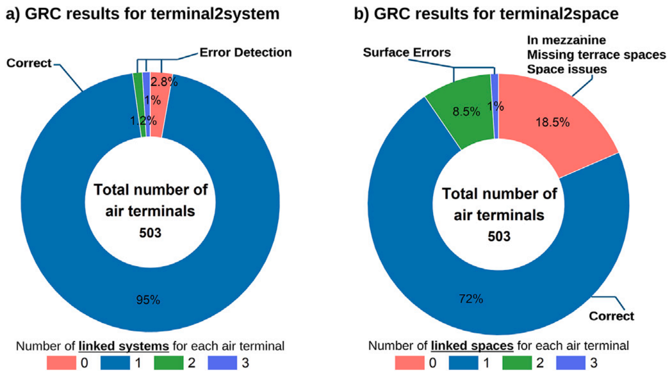

Fig. 14. Distribution of air terminals linked to: (a) systems and (b) spaces.

However, certain potential and elusive issues cannot be directly identified using the aforementioned rules. Therefore, this study introduces the path-finding approach, using depth-first search algorithms to further detect such issues. This approach should be implemented after the initial establishment of the MEP graphs. Fig. 14 illustrates the accuracy of the air-loop MEP knowledge graph established through this GRC-based workflow, which includes the application of initial error detection and fixes based on the aforementioned rules, and a final path-finding process. The results of the overall method are displayed in parts a and b in Fig. 14. Part (a) reveals that over 95% of the air terminals (Diffusers/grilles) were correctly linked to their corresponding systems (AHU/FCU/MVHR). About 2.2% of the air terminals were linked to two or three systems, with some supplied by FCUs linked to an MVHR in series, which is not a common design approach. A few of these instances were attributed to checking issues arising from adjacent duct entities that are not physically connected. Furthermore, the remaining 2.8% were not linked to any systems due to irregular human design errors in ductwork and relevant accessories. These results were verified using the building's HVAC schematic drawings. Through manual intervention for error resolution, all air terminals can achieve the correct systems using the path-finding approach, details of which are available in the following subsection.

Part (b) of Fig. 14 illustrates the precision of the link (containment relation) between spaces and air terminals. About 70% of the air terminals are accurately matched with their designated spaces. However, around 18.5% remain unassociated with any space due to certain spaces not extending high enough to reach the ceiling surfaces and several missing spaces on the terrace. Additionally, surface errors in air terminals were identified during the execution of the GRC tool, triggering the activation of bounding-box checking that resulted in the creation of a few non-existent links. Consequently, manual correction was required to eliminate the nonexistent links, affecting approximately 9.5% of the air terminals, which were erroneously connected to two or more spaces.

5.3.2. Geometry-induced MEP knowledge graph

After completing the error resolution process, a thorough and precise MEP knowledge graph is constructed, containing the interconnections among MEP elements. This graph is defined as the Geometry-induced MEP Knowledge Graph. To delve into the structural aspects of the generated graph, the following paragraphs analyze parts of the graph related to three data domains: air loops, HVAC-related water loops, and non-HVAC water loops.

5.3.2.1. MEP knowledge graph for air loops. Utilizing path-finding queries within the established Geometry-induced MEP knowledge graph facilitates the extraction of structural features and linkage information within MEP systems. Fig. 15 visually showcases the outcomes of querying air loops in BlenderBIM. The components involved, such as AHUs, their associated air terminals, and relevant ducts, were extracted to form their ventilation subsystems. Notably, many ventilation subsystems span multiple floors and areas, incorporating Variable Air Volume (VAV) boxes. Additionally, over fifty ventilation subsystems, driven by MVHRs and FCUs, were identified and extracted, although they are not visualized due to their simpler and smaller-scale structures. Compared to manual methods, graph-based queries expedite information extraction and system partitioning. This capability enhances the structural information integrated into the building management system and proves advantageous in identifying faults during operational issues.

5.3.2.2. MEP knowledge graph for HVAC-related water loops. Path-finding queries in the generated Geometry-induced MEP knowledge graph can also reveal HVAC-related water loop components and their networks. In the considered building example, the OPS podium is serviced by an air-cooled chiller and district heating systems via main water pipes originating from a service building (energy station). Consequently, this study initiated queries using the connection points between the primary water loop and secondary water loops as starting points, deviating from the use of energy source components (e.g., HIUs or Chillers). Fig. 16 visually demonstrates‘ the MEP networks formed from the graph queries, with the related water-loop IFC model split into two distinct domains: Low-Temperature Hot Water (LTHW) and Chilled Water (CHW). The analysis reveals that the LTHW system comprises only one secondary loop, while the CHW system incorporates two. Half of the AHUs are connected to both water loops, while the remaining half exclusively links to the LTHW loop. Additionally, all FCUs are connected to the CHW system, with 83% of them also having connections to the LTHW loop. This configuration aligns with the schematic drawings.

5.3.2.3. MEP knowledge graph for non-HVAC water loops. Non-HVAC water loop networks including domestic water pipework, rainwater pipework, and drainage pipework were included in the input IFC model of the GRC process. By employing a query-based extraction process, similar to HVAC-related water loops, three water loops were formed that include: a domestic hot water loop, a domestic cold water loop, and a rainwater and drainage pipework, as shown in Fig. 17. Unlike the HVAC domain, most terminals in these domestic cold/hot water loops (such as water taps or showers) were absent in the raw data. Consequently, despite the detection of numerous errors through the aforementioned rules, these discrepancies can be overlooked without necessitating revisions in the geometry-induced MEP knowledge graph. Furthermore, if an MEP element is not associated with any of the aforementioned loops, it can be classified as rainwater or drainage pipework. This is because these domains lack a clear loop structure, and such information is far less crucial than in the HVAC domain when considering building energy modeling.

Ultimately, the establishment of a geometry-induced MEP knowledge graph and the application of queries have led to a clear and structured representation of MEP systems. This approach holds the potential to significantly facilitate the development of building digital twins and enhance building management systems.

5.3.3. Breakdown of GRC results and graph evolution

The diagram in Fig. 18 showcases the distribution of connectivity pairs across three IFC files, originating from four stages within the GRC-based workflow: BIM context, GRC tool, BBOX, and manual intervention. Notably, the number of pairs from the BIM context in air loops is notably lower compared to water loops, attributed to MEP fabrication elements lacking semantic linking. This emphasizes the pivotal role of the GRC tool in identifying air-loop connections. Bounding-box checking remains vital, as it identifies roughly 40% connectivity pairs due to issues like non-face-to-face connections and surface errors. Concerning HVAC-related water loops, the BIM context contributes nearly 80% connectivity pairs, with 13% from the GRC tool and less than 9% from bounding-box checking, mainly due to unsupported geometric

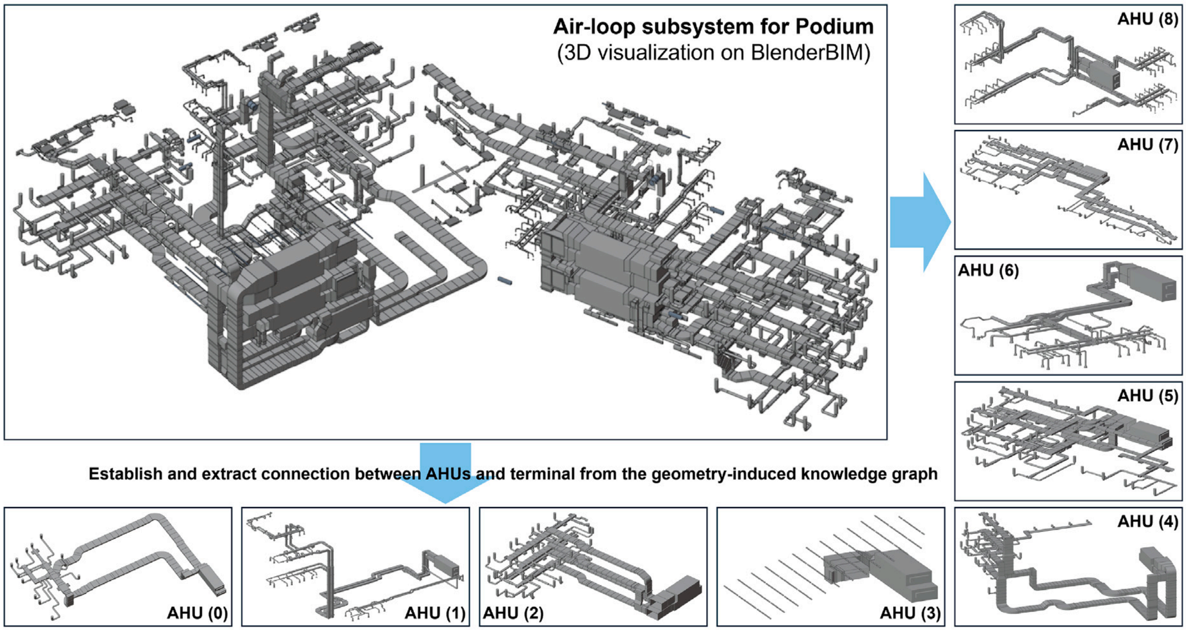

Fig. 15. Extraction of connection elements among AHUs and their air terminals, from the full air-loop IFC data set.

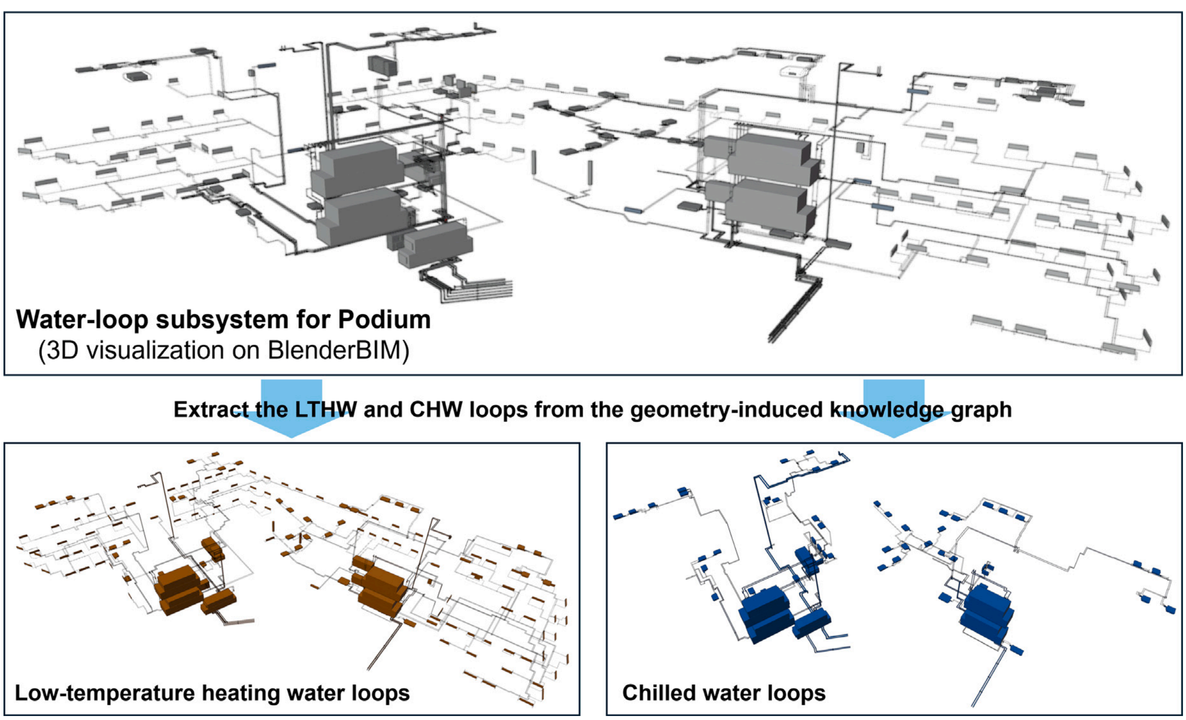

Fig. 16. Extraction of LTHW loop and CHW loop connection elements, from the full HVAC-related and water-loop IFC data set.

representations and human errors near AHUs/FCUs. Furthermore, the combination of the BIM context and the GRC tool effectively manages elements in non-HVAC water loops, given their simpler geometric representations.

Fig. 19 presents geometry-induced MEP knowledge graphs for HVAC-related air and water loops, illustrating their progressive enrichment at each stage. In the enrichment process for the HVAC air loops (top row), BIM designers employed fabrication duct elements, assembling them into ductwork without semantic links, resulting in a very sparse initial graph with few linked entities. Introducing the GRC tool notably improved ductwork network completeness, filling numerous missing links. Subsequently, BIM elements with surface errors were checked using bounding boxes, further enhancing graph completeness. A ruleset check corrected certain problematic entities, completing the graph establishment. The number of links detected at each stage can be found in Fig. 18. Unlike the air loop, the HVAC water loop graph (bottom row) showed more complete BIM semantic links due to fewer fabrication pipes, allowing most link relationships to be directly extracted. Enrichment through GRC and bounding boxes provided a fully comprehensive graph, as shown in the final graph, demonstrating that the proportion of detected links is related to BIM data quality.

Overall, achieving full automation of geometric relation checking poses challenges due to the varied nature of MEP-BIM files and their inconsistent data quality. The proposed GRC-based workflow addresses

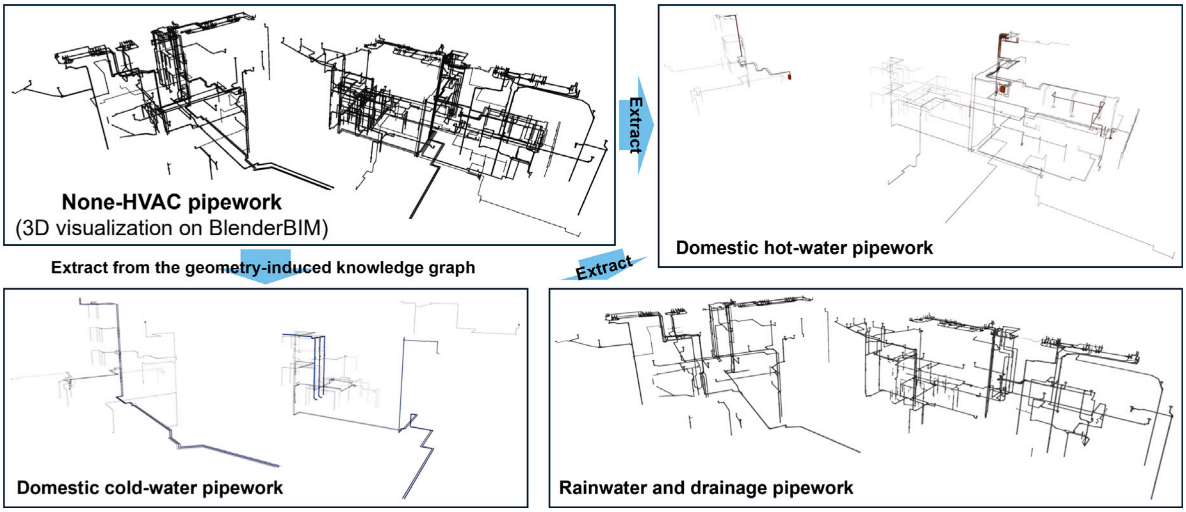

Fig. 17. Extraction of domestic hot/cold water loop, rainwater and drainage pipework elements from the full whole non-HVAC and water-loop IFC data set.

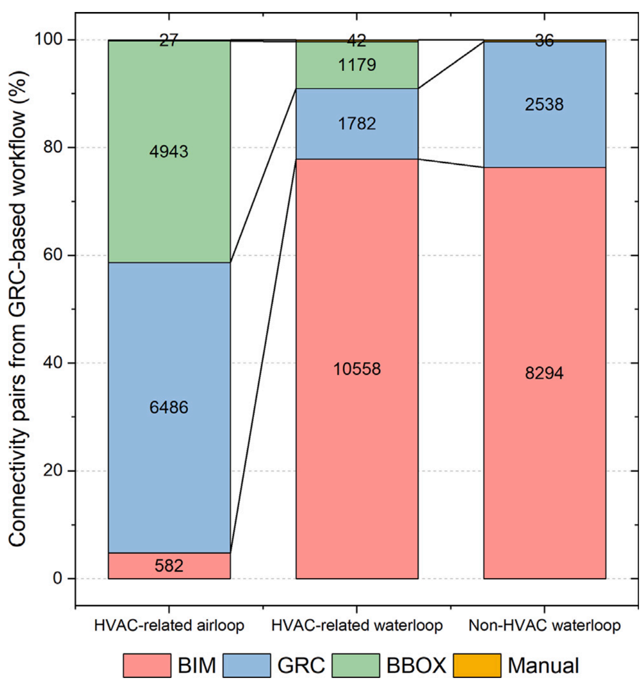

Fig. 18. Distribution of the connectivity pairs obtained from different stages in the GRC-based workflow.

this challenge by pinpointing errors/issues, significantly speeding up the establishment of the knowledge graph, and ensuring its overall completeness.

6. Discussion

The proposed geometric methodology is applied to MEP HVAC and ARC BIM data with the goal of identifying connections between HVAC source and sink devices, as well as determining the locations of HVAC terminal devices within the building's spatial volumes. The main objective of this approach is to create an HVAC connection graph that facilitates the automatic generation of building energy performance simulation models. This work is differentiated from existing methods in the literature, which focus on clash detection and adjacency identification for BIM quality checks and code compliance.

Additionally, as demonstrated in the large and complex application example involving HVAC networks for air, hot water, and cold water transfer in the podium of UCL's One Pool Street building, the proposed methodology successfully infers connections between HVAC source and sink components, as well as containments between HVAC terminal units and building spaces, in a relatively short time (a few hours). This relatively short execution time, is achieved despite the need to evaluate millions of element pairs for thousands of components, all without any prior knowledge of any connections.

GRC is domain-agnostic and can be applied to geometric data from various fields, such as architecture. For instance, adjacent surfaces between building walls and slabs can reveal thermal bridges, which are critical for building energy performance simulations to assess thermal losses. Furthermore, GRC can be utilized across different domains, as demonstrated in this study with ARC and MEP data. This capability is especially valuable for breaking down siloed data approaches, allowing for cross-domain semantic knowledge graph enrichment by integrating data from multiple fields and will be exploited in future developments.

7. Limitations and future directions

Although the proposed process successfully identified most HVAC element connections and containments among the HVAC terminal units and building spaces, it has several limitations, primarily due to the quality of the BIM input data.

If surfaces in both boundary representations of the BIM elements of a checked pair are missing or incorrectly oriented – facing inward instead of outward – clash and containment detection for that pair cannot be performed. In such cases, regions of the three-dimensional space cannot be classified as inside or outside any of the elements in the pair. Only pairs in which neither element or only one element has a surface error can be effectively checked for clashes or containments.

If BIM elements are attached or intersecting but not actually connected (as shown in part a of Fig. 13 and the top right case in Table 1), are mistakenly identified as connected by the proposed methodology. These cases are characterized as False Positive (FP) cases. Similarly, there are instances where MEP components are physically connected in reality, but are not geometrically connected in the as-designed BIM data (as illustrated in parts b and c of Fig. 13 and in the bottom left case in Table 1). These cases are classified as False Negative (FN) cases. To correct these inconsistencies, the methodology will be enhanced with a GUI-assisted deletion process in the future, allowing the user to detect, visually verify and address such cases via the use of an appropriate BIM authoring tool.

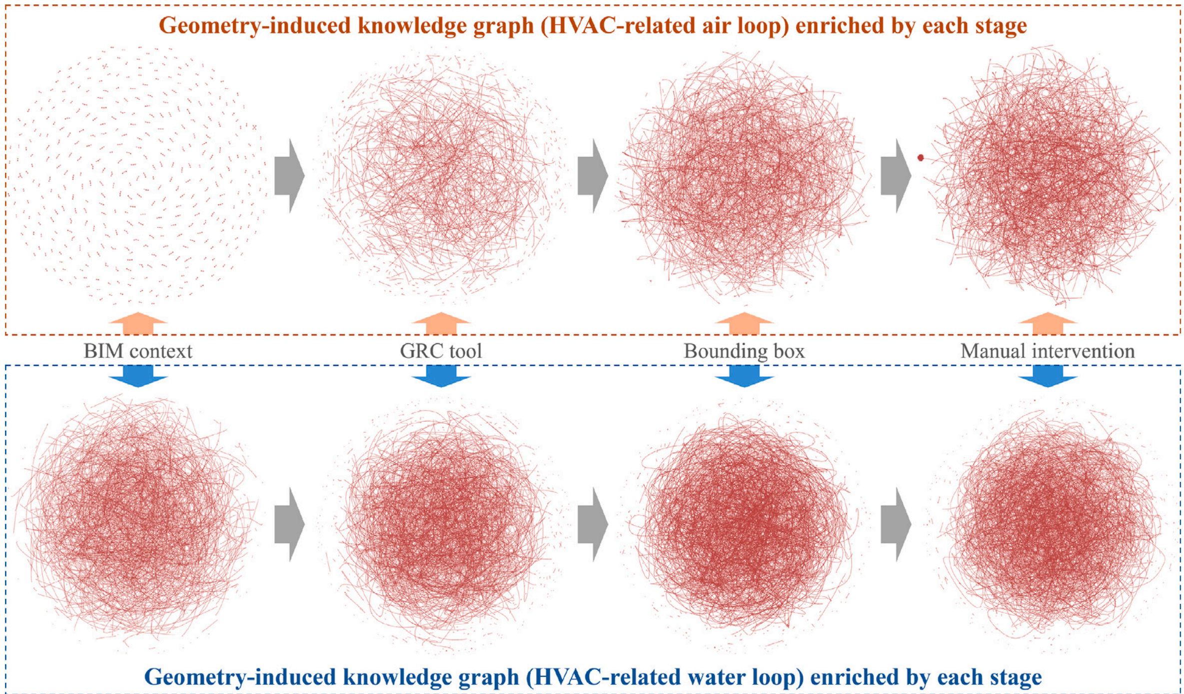

Fig. 19. Visualization of the geometry-induced MEP knowledge graphs enriched through the GRC-based workflow.

The design errors presented in Fig. 13 result in missing or incorrect connections between BIM elements, which cause inaccuracies in the generated HVAC topology graph. These errors can be corrected in future developments, through an iterative process involving visual inspection and verification using a GUI, followed by design corrections and IFC export using a BIM authoring tool. By the end of this process, an error-free BIM MEP file will be produced, enabling the proposed methodology to generate an accurate HVAC topology that is ready for building energy performance simulation.

Finally, in some cases, the semantic connections inferred from geometric relation checks between pairs of BIM elements are already present in the BIM data. In these situations, the proposed methodology can be used as a supplementary tool to validate the accuracy of these pre-existing semantic connections. The geometry-induced MEP graphs have the potential to generate a comprehensive HVAC system topology, supporting the automation of building performance simulation models and aiding fault detection and diagnostics during building operations through the development of building digital twins. The knowledge graph, as a digital counterpart that abstracts and represents real-world information, offers designers a more accessible understanding of the MEP systems. This structured representation can further facilitate the identification and correction of errors that might otherwise remain hidden in raw data.

In conclusion, the complexity of MEP systems makes the geometric relationships among MEP elements insufficient for accurately inferring semantic connections. This work proposes an a posteriori approach, using error detection rules combined with manual intervention, to identify and correct potential issues, thus establishing reliable semantic connections. However, directly inferring precise semantic relations from geometric data alone remains challenging. Future work will focus on refining the error detection ruleset, to enhance the reliability of inferring semantic connections from geometric relationships.

8. Conclusions

The geometric connections within BIM models are crucial for uncovering non-geometric semantic connections among their elements. Leveraging these connections can effectively deduce further semantic relationships among the elements. This dual function not only aids in connecting isolated structures within semantic graphs across different data domains but also enhances the connectivity of existing graphs by introducing additional links between elements.

As it was demonstrated, geometric containment relations between IFC space elements from the architecture domain and MEP terminal units from the MEP domain can be used to infer containment relations among these elements, supporting a cross-domain linkage between architecture and MEP. Additionally, intersections or adjacencies between the geometric representations of MEP elements might reveal flow connections between them. The exploitation of this concept has two benefits. On the one hand, the identification of these semantic connections can help in the generation of the HVAC topology of a building and thus facilitate the automatic BIM2BEM generation process. On the other, these geometric relations can reveal design errors involving MEP components, such as clashes between pipe elements. Finally, using semantic reasoning on the detected geometric relationships, missing semantic connections that are related to missing MEP elements can also be identified.

To enable these operations, a Geometric Relation Checking (GRC) tool is introduced. GRC detects three types of geometric relations among pairs of geometric representations of BIM elements contained in IFC files that are clash, intersection, and containment. Using these relationships applied to the geometric representations of building space volumes and MEP elements, connectivity relations among MEP elements, as well as containment relations between MEP elements and building spaces, can be inferred. To enable fast execution, GRC is written in C++ using multi-thread libraries to enable parallel runs on multi-core processors achieving millions of pair checks in a minute time scale.

It has been shown that the GRC tool can serve as a central element within a process flow that produces a Geometry-induced knowledge graph representing a building's MEP (Mechanical, Electrical, Plumbing) network using architectural and MEP BIM (Building Information Modeling) input data in the IFC format. The effectiveness of this proposed workflow was demonstrated through its application to the air-loop and water-loop systems of the podium in UCL's One Pool Street building. The HVAC system of this building showcases complexity and diversity in MEP components, as well as substantial data capacity, providing sufficient evidence of the proposed workflow's capability in handling large and intricate datasets. As the demand for knowledge graph generation grows to support the digitalization of building stocks, the necessity for workflows like the one introduced will become increasingly evident, potentially expanding its utilization beyond architecture and MEP to other BIM data domains.

CRediT authorship contribution statement

Georgios Nektarios Lilis: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Meng Wang:** Writing – review & editing, Writing – original draft, Validation, Software, Conceptualization. **Kyriakos Katsigarakis:** Software, Methodology, Conceptualization. **Dimitrios Mavrokapnidis:** Software, Data curation. **Ivan Korolija:** Validation, Conceptualization. **Rovas Dimitrios:** Validation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Horizon Europe DigiBUILD project under contract No. 101069658, and co-funded by the UK Researchand Innovation (UKRI) under the UK government's Horizon Europe funding guarantee grant number 10040988.

Data availability

Data will be made available on request.

References

[1] P. Pauwels, A. Costin, M.H. Rasmussen, Knowledge graphs and linked data for the built environment, in: Industry 4.0 for the Built Environment: Methodologies, Technologies and Skills, Springer, 2021, pp. 157–183, http://dx.doi.org/10.1007/978-3-030-82430-3_7.

[2] ASHRAE, Designation and classification of semantic tags for building data, 2024, https://docs.open223.info/intro.html.

[3] I.E.A. EBC, Annex 81 data-driven smart buildings, 2024, https://annex81.iea-ebc.org/.

[4] K. Katsigarakis, G.N. Lilis, D. Rovas, A cloud IFC-based BIM platform for building energy performance simulation, in: Proceedings of European Conference on Computing in Construction, 2021, pp. 350–357, http://dx.doi.org/10.35490/EC3.2021.177.

[5] A. Desrochers, A. Clément, A dimensioning and tolerancing assistance model for CAD/CAM systems, Int. J. Adv. Manuf. Technol. 9 (1994) 352–361, http://dx.doi.org/10.1007/BF01748479.

[6] T.-H. Nguyen, A.A. Oloufa, K. Nassar, Algorithms for automated deduction of topological information, Autom. Constr. 14 (1) (2005) 59–70, http://dx.doi.org/10.1016/j.autcon.2004.07.015.

[7] W. Solihin, C. Eastman, Classification of rules for automated BIM rule checking development, Autom. Constr. 53 (2015) 69–82, http://dx.doi.org/10.1016/j.autcon.2015.03.003.

[8] P. Demian, K. Ruikar, T. Sahu, A. Morris, Three-Dimensional Information Retrieval (3DIR): exploiting 3D geometry and model topology in information retrieval from BIM environments, Int. J. 3-D Inf. Model. 5 (1) (2016) 67–78, http://dx.doi.org/10.4018/978-1-5225-5191-1.ch043.

[9] L. Ma, R. Sacks, U. Kattel, T. Bloch, 3D object classification using geometric features and pairwise relationships, Computer-Aided Civ. Infrastruct. Eng. 33 (2) (2018) 152–164, http://dx.doi.org/10.1111/mice.12336.

[10] H. Luo, G. Gao, H. Huang, Z. Ke, C. Peng, M. Gu, A geometric-relational deep learning framework for BIM object classification, in: Proceedings of European Conference on Computer Vision, Springer, 2022, pp. 349–365, http://dx.doi.org/10.48550/arXiv.2212.00942.

[11] A. Borrmann, E. Rank, Topological analysis of 3D building models using a spatial query language, Adv. Eng. Inform. 23 (4) (2009) 370–385, http://dx.doi.org/10.1016/j.aei.2009.06.001.

[12] A. Khalili, D.H. Chua, IFC-based graph data model for topological queries on building elements, J. Comput. Civ. Eng. 29 (3) (2015) 04014046, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000331.

[13] C. van Treeck, E. Rank, Dimensional reduction of 3D building models using graph theory and its application in building energy simulation, Eng. Comput. 23 (2007) 109–122, http://dx.doi.org/10.1007/s00366-006-0053-7.

[14] J. Choi, J. Lee, 3D geo-network for agent-based building evacuation simulation, in: 3D Geo-Information Sciences, Springer, 2009, pp. 283–299, http://dx.doi.org/10.1007/978-3-540-87395-2_18.

[15] P. van den Helm, M. Böhms, L. van Berlo, IFC-based clash detection for the open-source BIMserver, in: Proceedings of Computing in Civil and Building Engineering International Conference, vol. 181, Nottingham University Press, 2019.

[16] A.O. Akponeware, Z.A. Adamu, Clash detection or clash avoidance? An investigation into coordination problems in 3D BIM, Buildings 7 (3) (2017) 75–103, http://dx.doi.org/10.3390/buildings7030075.

[17] J.-H. Seo, B.-R. Lee, J.-H. Kim, J.-J. Kim, Collaborative process to facilitate BIM-based clash detection tasks for enhancing constructability, J. Korea Inst. Build. Constr. 12 (3) (2012) 299–314, http://dx.doi.org/10.5345/JKIBC.2012.12.3.299.

[18] R. Chahrour, M.A. Hafeez, A.M. Ahmad, H.I. Sulieman, H. Dawood, S. Rodriguez-Trejo, M. Kassem, K.K. Naji, N. Dawood, Cost-benefit analysis of BIM-enabled design clash detection and resolution, Constr. Manag. Econ. 39 (1) (2021) 55–72, http://dx.doi.org/10.1080/01446193.2020.1802768.

[19] Y. Hu, D. Castro-Lacouture, C.M. Eastman, Holistic clash detection improvement using a component dependent network in BIM projects, Autom. Constr. 105 (2019) 102832, http://dx.doi.org/10.1016/j.autcon.2019.102832.

[20] J. Zhang, Z. Hu, BIM-and 4D-based integrated solution of analysis and management for conflicts and structural safety problems during construction: 1. Principles and methodologies, Autom. Constr. 20 (2) (2011) 155–166, http://dx.doi.org/10.1016/j.autcon.2010.09.013.

[21] A.S. Ismail, K.N. Ali, N.A. Iahad, A review on BIM-based automated code compliance checking system, in: International Conference on Research and Innovation in Information Systems, IEEE, 2017, pp. 1–6, http://dx.doi.org/10.1109/ICRIIS.2017.8002486.

[22] V. Getuli, S.M. Ventura, P. Capone, A.L. Ciribini, BIM-based code checking for construction health and safety, Procedia Eng. 196 (2017) 454–461, http://dx.doi.org/10.1016/j.proeng.2017.07.224.

[23] Y. Wang, Y. Liu, H. Cai, J. Wang, X. Zhou, An automated fire code compliance checking jointly using building information models and natural language processing, Fire 6 (9) (2023) 358, http://dx.doi.org/10.3390/fire6090358.

[24] X. Zhao, L. Huang, Z. Sun, X. Fan, M. Zhang, Compliance checking on topological spatial relationships of building elements based on building information models and ontology, Sustainability 15 (14) (2023) 10901, http://dx.doi.org/10.3390/buildings7030075.

[25] Solibri, Solibri model checker, 2014, https://www.solibri.com/.

[26] novaCITYNETS, FORNAX cloud, 2017, https://fornaxcloud.com/.

[27] J.-K. Lee, C.M. Eastman, Y.C. Lee, Implementation of a BIM domain-specific language for the building environment rule and analysis, J. Intell. Robot. Syst. 79 (2015) 507–522, http://dx.doi.org/10.1007/s10846-014-0117-7.

[28] N. Kayhani, B. McCabe, B. Sankaran, Semantic-aware quality assessment of building elements using graph neural networks, Autom. Constr. 155 (2023) 105054, http://dx.doi.org/10.1016/j.autcon.2023.105054.

[29] M. Bueno, F. Bosché, Pre-processing and analysis of building information models for automated geometric quality control, Autom. Constr. 165 (2024) 105557, http://dx.doi.org/10.1016/j.autcon.2024.105557.

[30] J. Zhu, P. Wu, X. Lei, IFC-graph for facilitating building information access and query, Autom. Constr. 148 (2023) 104778, http://dx.doi.org/10.1016/j.autcon.2023.104778.

[31] P. Pauwels, ifcOWL, 2019, https://technical.buildingsmart.org/standards/ifc/ifc-formats/ifcowl/.

[32] M. Bondue, A. Wagner, P. Pauwels, FOG: File ontology for geometry formats, 2020, https://mathib.github.io/fog-ontology/.

[33] K. McGlinn, A. Wagner, P. Pauwels, P. Bonsma, P. Kelly, D. O'Sullivan, Interlinking geospatial and building geometry with existing and developing standards on the web, Autom. Constr. 103 (2019) 235–250, http://dx.doi.org/10.1016/j.autcon.2018.12.026.

[34] N. Abadie, G. Atemezing, Ontology of geometric primitives, 2019, http://data.ign.fr/def/geometrie/20190212.en.htm.

[35] A. Wagner, M. Bonduel, P. Pauwels, R. Uwe, Relating geometry descriptions to its derivatives on the web, in: Proceedings of the European Conference on Computing in Construction, European Council on Computing in Construction, 2019, pp. 304–313, http://dx.doi.org/10.35490/EC3.2019.146.

[36] Z. Wang, B. Ouyang, R. Sacks, Graph-based inter-domain consistency maintenance for BIM models, Autom. Constr. 154 (2023) 104979, http://dx.doi.org/10.1016/j.autcon.2023.104979.

[37] B. Ouyang, Z. Wang, R. Sacks, Semantic enrichment of object associations across federated BIM semantic graphs in a common data environment, in: ECPPM 2022-EWork and EBusiness in Architecture, Engineering and Construction 2022, CRC Press, 2023, pp. 591–598.

[38] M. Li, Z. Wang, G. Fierro, C.H.C. Man, P.M.P. So, K.F.C. Leung, Developing an automatic integration approach to generate brick model from imperfect building information modelling, J. Build. Eng. 97 (2024) 110697, http://dx.doi.org/10.1016/j.jobe.2024.110697.

[39] Brick Consortium, Brick: A uniform metadata schema for buildings, 2023, https://brickschema.org/.

[40] G.N. Lilis, M. Wang, K. Katsigarakis, D. Mavrokapnidis, I. Korolija, D. Rovas, Clash, Containment and Adjacency Detection Functions on Oriented Polygon Sets Using Binary Space Partitioning Trees, Zenodo, 2024, http://dx.doi.org/10.5281/zenodo.14039972.

[41] G.N. Lilis, G. Giannakis, D. Rovas, Automatic generation of second-level space boundary topology from IFC geometry inputs, Autom. Constr. 76 (2017) 108–124, http://dx.doi.org/10.1016/j.autcon.2016.08.044.

[42] B. Vatti, A generic solution to polygon clipping, Commun. ACM 35 (7) (1992) 56–63, http://dx.doi.org/10.1145/129902.12990.

[43] V. Kukkonen, A. Kücükavci, M. Seidenschnur, M.H. Rasmussen, K.M. Smith, C.A. Hviid, An ontology to support flow system descriptions from design to operation of buildings, Autom. Constr. 134 (2022) 104067, http://dx.doi.org/10.1016/j.autcon.2021.104067.

[44] D. Mavrokapnidis, G.N. Lilis, K. Katsigarakis, I. Korolija, D. Rovas, Semi-automated extraction of HVAC system topology from imperfect Building Information Models, in: Proceedings of the 18th Conference of the International Building Performance Simulation Association, IBPSA, Shanghai, China, 2023, pp. 963–970, http://dx.doi.org/10.26868/25222708.2023.1392.

[45] M. Wang, G.N. Lilis, D. Mavrokapnidis, K. Katsigarakis, I. Korolija, D. Rovas, A knowledge graph-based framework to automate the generation of building energy models using geometric relation checking and HVAC topology establishment, Energy Build. 325 (2024) 115035, http://dx.doi.org/10.1016/j.enbuild.2024.115035.

[46] Y.-Q. Xiao, S.-W. Li, Z.-Z. Hu, Automatically generating a MEP logic chain from building information models with identification rules, Appl. Sci. 9 (11) (2019) 2204, http://dx.doi.org/10.3390/app9112204.

[47] Y. Wang, L. Zhang, H. Yu, R.L. Tiong, Detecting logical relationships in mechanical, electrical, and plumbing (MEP) systems with BIM using graph matching, Adv. Eng. Inform. 54 (2022) 101770, http://dx.doi.org/10.1016/j.aei.2022.101770.