Automation in Construction 166 (2024) 105628

Contents lists available at ScienceDirect

Automation in Construction

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

Topological BIM for building performance management

Angelo Massafra [a,*], Wassim Jabi [b], Riccardo Gulli [a]

[a] Department of Architecture, Alma Mater Studiorum University of Bologna, Italy
[b] Welsh School of Architecture, Cardiff University, UK

ARTICLE INFO

Keywords:
Building management
Building topology
Building information modeling
Building performance simulation
Building energy modeling

ABSTRACT

Despite the recognized benefits of Building Information Modeling (BIM) and Building Performance Simulation (BPS), these procedures are often time- and resource-intensive, posing significant barriers to adoption in building management. While significant advancements have been made in automating BIM generation using scanning technologies, these applications often yield geometrically complex, product-oriented, and semantic-poor models, often incompatible with BPS tools, which are instead space-oriented. This paper introduces a semi-automated method for generating space-oriented BIM models tailored for BPS. The process utilizes topological and conditional modeling principles to create semantically defined, and information-rich models suitable for simulation environments called Topological BIM (TBIM). This method ensures semantic standardization, rapid digitization, and high interoperability, facilitating progressive data enrichment for building digital models. In the paper, a case study of a higher education building is presented to demonstrate the approach and validate a toolkit developed for delivering the TBIM models.

1. Introduction

Given the vast supply of existing buildings serving our everyday activities and the current energy challenges, there is an immediate need to improve built heritage performance [1]. This challenge is particularly relevant for the administrators of large building stocks, especially those in the public sector. In Italy, for instance, local public administrations own about 10% of the national built heritage, of which approximately 48% consists of structures constructed more than 40 years ago [2] without specific standards on building performance efficiency [3]. Acknowledged the economic and technical impracticality of extensive renovations or replacements of such stocks, minimally invasive methods are needed to improve operational performance and support management processes in the short term [4].

Within this context, the digitization of built asset management has become imperative to assist building administrators in managing buildings and orienting building operations towards a performance-oriented perspective [5]. Due to the digitization needs in the AECO (Architecture, Engineering, Construction and Operation) sector, many information technologies have emerged in recent decades to enable new ways of capitalizing building knowledge towards cost-benefit optimization, decision-making and strategic planning [6]. These include Building Information Modeling (BIM) [7], Heritage BIM [8], Building Performance Simulation (BPS) [9], Smart Buildings [10], Digital Twin (DT) [11] and Artificial Intelligence (AI) [12]. Among them, BIM is considered central for delivering tools suitable for building management, consisting of a ‘skeleton’ for semantically structuring all the data related to the building lifecycle [13,14]. Similar considerations can be made concerning BPS, a technology widely used to assess building performance [15].

Despite BIM and BPS’s advantages, many limitations exist to their full adoption within building management processes [16]. On the one hand, BIM modeling procedures are time- and resource-intensive [17]. Although considerable progress has been achieved in automatically creating geometrically intricate models to speed up BIM, the resulting models are often excessively large, require significant computational resources, and are rarely suitable for performance analysis uses [18]. On the other hand, numerous challenges persist in achieving interoperability between BIM and BPS systems, often difficult to overcome due to the differences in how these systems approach data and geometry modeling [19]. Indeed, since its inception, BIM has focused on modeling the physical components of buildings adopting a product-oriented

Abbreviations: DT, Digital Twin; BEM, Building Energy Modeling; BIM, Building Information Modeling; BPS, Building Performance Simulation; ILD, Informational Load Dictionary; IRS, Informational Ruleset; TBIM, Topological Building Information Modeling.

* Corresponding author at: Viale del Risorgimento 2, Bologna 40136, Italy.
E-mail address: angelo.massafra2@unibo.it (A. Massafra).

https://doi.org/10.1016/j.autcon.2024.105628
Received 11 December 2023; Received in revised form 10 July 2024; Accepted 10 July 2024
Available online 14 July 2024

perspective that organizes building data within volume-based informational objects [20]. In contrast, BPS has always adopted a space-oriented approach, decomposing buildings into spatial elements and representing them through planar surfaces [21]. This divergence, which extends to software compatibility challenges [22], usually results in inefficient or erroneous data exchange between BIM and BPS. As a result, it is common practice for BPS models to be reconstructed from scratch each time they are needed for analysis, a practice in contrast with the potential efficiency of using a single model – the BIM – to both organize building knowledge and conduct performance analysis.

Acknowledged these limitations, this paper aims to:

1) Introduce a methodology for the semi-automated generation of BPS-compatible BIM models for building performance management. This methodology, which aims to shift the BIM modeling approach from product-oriented to space-oriented, is grounded on topological and conditional modeling principles that allow for creating lightweight, semantically consistent, topologically connected, and information-rich conceptual information models.
2) Define the 'Topological BIM' (TBIM) approach, contrasting it with traditional product-oriented BIM. TBIM represents buildings as conceptual models resulting from aggregating spatial elements rather than construction components. These models are simple from a geometric standpoint but rich in semantic information, making them well-suited for BPS applications. Their conceptual nature and the topological interconnection of the elements within them provide a useful informational basis for deriving more geometrically detailed models, if necessary.
3) Present a toolkit developed for applying the TBIM methodology and demonstrate its application in a significant case study. This toolkit leverages the Topologicpy software library [23] as the core for managing the topological, geometrical, and informational aspects of TBIM models and uses the LadyBug tools [24] in combination with the EnergyPlus calculation engine [25] to conduct energy performance simulations.

The implications of this study in the broader research context include overcoming some of the barriers associated with BIM and BPS in building management. Proposing a topological and space-oriented approach to BIM, the research aims to reduce the high time and costs usually associated with BIM development by semiautomating the modeling tasks, promoting semantic standardization of BIM procedures by framing them in a solid theoretical framework, and improving interoperability between information modeling and performance analysis tools by facing the BIM-BPS integration issue in the early stages of BIM modeling. The added value of the study lies in introducing and demonstrating the so-called TBIM concept, which, although implicitly present in current research on BPS, lacks both theoretical formalization and practical application in the BIM field.

The paper is structured as follows. Section 2 presents the state of the art of the topic. Section 3 provides the theoretical framework and the methodological background of the research. Section 4 illustrates the application of the method to a case study building through the developed toolkit. Finally, Section 0 presents the results and limitations of the proposed approach and suggests further improvements. A GitHub repository is available to inspect the materials and models presented in the text [26].

2. State of the art

2.1. Automated generation of BIM for existing buildings

Manual BIM requires significant time and investment, which pose significant barriers to adopting this technology for large building portfolios. While the traditional approach is valuable for developing geometrically detailed models of building physical components, the resultant models can be large and computationally expensive to process and maintain for administrations. Furthermore, since different modelers could be involved in digitizing portfolios, manually created models can lead to discrepancies and semantic inconsistencies across models of different buildings. For this reason, computer-aided generation of BIM for existing buildings has been investigated in recent years [27]. Three major research areas have mainly emerged: (1) automated point cloud semantic segmentation, labeling, and tagging; (2) automated conversion of 3D point clouds to BIMs, also referred to as 'scan-to-BIM'; and (3) automated conversion of 2D drawings to BIM.

2.1.1. Automated point cloud semantic segmentation, labeling, and tagging

3D point cloud semantic segmentation, labeling, and tagging aim to enrich point clouds – which are 3D models derived from photogrammetric and laser scanning surveys containing exclusively dimensional data – with information that can be extracted from the context on a semantic basis. The association of information to the points usually occurs by assigning labels to the points that identify classes or types of construction elements, as well as properties derivable from the geometry of the objects. These methods have led to many applications, mainly in building cultural heritage. For instance, Romero-Jarèn and Arranz found a method for automatically segmenting and labeling point clouds in building elements, encompassing floors, ceilings, walls, and columns [18]. Weinmann et al. introduced a method for the semantic interpretation of point clouds based on supervised machine learning (ML) techniques [28]. Valero et al. developed a method for automating the detection of defects in masonry walls in TLS point clouds [29]. Hsieh et al. investigated the automated semantic segmentation of indoor point clouds derived from close-range images using 3D deep learning [30]. Most of these research efforts aim to extract and model semantic information about construction components rather than spaces, leading to product-based BIM models rarely being finalized for BPS purposes.

2.1.2. Scan to BIM

Scan-to-BIM involves automated or manual techniques for generating BIM models from 3D scans. The process starts with segmenting point clouds into groups and defining these groups semantically through class assignments. Subsequently, algorithms are applied to the point cloud to transform point groups into informational objects capable of storing data. The scan-to-BIM techniques can be numerous, ranging from simple automation scripts to visual programming- (VP) [31] and even AI-based workflows [32]. Among the many studies in the literature, Boschè et al. explored the benefits of using scan-to-BIM for MEP components [33]. Lee et al. introduced a graph-based deep learning model to represent 3D objects, such as bridge components, within a BIM framework [34]. In addition, Yin et al. proposed a deep learning-based approach to automatically generate as-built BIM from point clouds within industrial facilities [35]. In these cases, the advantages offered by automation are significant. However, the scan-to-BIM research field appears to be more focused on the 3D modeling aspects of BIM rather than performance information, resulting in models that are very complex geometrically and heavy to manage, often unusable in BPS software.

2.1.3. 2D to BIM

The 2D to BIM approach contrasts these methods [36]. This approach involves leveraging existing building documentation, such as 2D-floor plans, sections, and elevations in CAD and PDF files to enable the automated generation of BIM models. Although these models are more simplified than those produced through manual or scan-to-BIM methods, they are lighter and easier to manage. Among the literature is the study by Bortoluzzi et al., who developed an automated process that uses 2D floorplans and elevation drawings to generate semantic BIMs rich in information for facility management (FM) [17]. Lu et al. generated Industry Foundation Class (IFC) models from 2D drawings and further attached material information on components through onsite surveying [37]. Yang et al., instead, explored 3D BIM modeling from 2D CAD drawings, focusing on structural components [38]. Despite their scientific merit, these research efforts are also directed towards a product-oriented view of construction. When aimed at space management and designed for use in BPS environments, such as the research by Bortoluzzi et al. [17], the models produced, while compatible with BPS, are not filled with valuable information for performance analysis or no demonstration in BPS environments is provided.

2.2. Building topology modeling

Topological modeling of buildings can help integrate the product-oriented with the space-oriented view of BIM, which is essential for orienting BIM towards BPS uses. On the one hand, topology modeling can allow for structuring building information around spatial elements. On the other hand, it can allow for the representation of the interface components that directly affect building performance (such as partition and opening components) and connect them to the spatial elements they bind. Indeed, according to most architectural topological conceptions, a building can be viewed as a collection of spatial elements that aggregate and relate to each other through containment, adjacency, and passage relationships. Bounded by interface elements (e.g., walls, floors and roofs), these spatial elements can be represented as objects in a schematic form and characterized by relationships and attributes.

2.2.1. Data-driven approaches for modeling building topology

Data-driven approaches to building topology modeling rely on Linked Building Data (LBD). LBD utilizes data modeling methods based on ontologies for modeling building topology [39]. Examples include IFC, BOT (Building Topology Ontology), and Brick. IFC is the international open BIM standard reference [40]. It organizes spatial elements according to a spatial hierarchy, including elements such as IfcSpace, IfcZone, IfcBuildingStorey, IfcBuilding, and IfcSite, interconnected through the IfcLocalPlacement relationship. BOT is a streamlined ontology focused solely on fundamental concepts of building topology, including physical and conceptual components and their connections. Developed by the World Wide Web Consortium (W3C), BOT offers a simplified representation compared to IFC by modeling spatial elements as 'Zones' [41]. It also facilitates the modeling of partition elements as 'Interfaces' and specifies the relationships among spatial, interface, and generic elements. Brick, instead, is an open-source schema developed for smart buildings, distinguished by its practicality in mapping the topological relationships between spatial elements (encoded in Brick as the 'Location' class) and system components (encoded as 'Equipment') [42]. It offers a more simplified approach than IFC, which is particularly beneficial when 3D geometries of system components are absent [14].

2.2.2. Model-based approaches for topology modeling

While IFC is both an ontology and a data format, Brick and BOT do not inherently offer BIM and BPS modeling tools within 3D environments. However, they serve as foundational ontologies upon which such tools can be built. Integrating these ontologies with established modeling tools and formats in the BIM realm can enable the execution of semantically structured and automated BIM, for instance, by implementing conditional modeling techniques based on topology-related concepts. Examples of this approach are the works by Postle [43] and Villegas-Ballesta [44], both based on Topologic, a open-source software designed explicitly for the topological modeling of buildings [45]. Postle's work involved the creation of a tool to carry out BIM modeling by transforming simple 3D models into IFC models through rule-based specifications on building topology and space functions. Villegas-Ballesta's study focuses instead on conceptualizing a topology-based knowledge model to be applied to architectural design to enrich digital models with semantic information. The method proposed by this author allows for the semi-automated generation of BIM models thanks to VP scripts by leveraging the topological and functional properties of spaces and principles of conditional modeling. Along with that of Janssen et al. [46], these works are applied to the early-stage design of new buildings rather than the management of existing buildings. However, they are all framed within a similar theoretical framework that has contributed to shaping the background of this research.

2.3. BIM-based BPS

As BIM has grown in popularity, BPS has begun to be incorporated into design and management processes as a part of more intricate cross-disciplinary models [47]. However, bidirectional connectivity between BPS and BIM technologies is still lacking [19]. To conduct performance simulations, current procedures frequently require starting from the BIM and then building a second unlinked digital model for performance analysis. Splitting models causes information loss, time-consuming and repeated procedures, and, sometimes, misunderstandings between managers and performance analysts.

2.3.1. Gaps and challenges to BIM-to-BEM integration

The interoperability problem between BIM and BPS, particularly Building Energy Modeling (BEM), has been recently investigated by several studies. According to these, there are still a lot of unsolved problems in the development of BIM-based BEM [48]. The first gap is that, due to the complexity of geometry, BIM-based energy simulations may cause processing bottlenecks. Since BPS tools typically demand models with regular squared mesh or surfaces, high polygon counts in BIM may lead to longer and uncontrollable simulation runs [49]. Moreover, BIM and BEM systems may employ different geometry kernels, negatively impacting their integration [50]. Another gap is related to the typically available data formats for BIM and BEM: IFC and Green Building Extensible Markup Language (gbXML) [51]. gbXML is a data format that stores almost all the building data required for BEM. It is based on planar-shaped surfaces rather than 3D objects, as instead IFC. Moreover, not all the properties codified by IFC are transferable to gbXML or vice versa.

2.3.2. Approaches enhancing BIM-to-BEM interoperability

Various approaches have been attempted to fill the interoperability gaps. For example, some scholarly investigations have advanced BIM-to-BEM interoperability by optimizing manual 'export-import' protocols in commercially available tools [22,52,53]. Other researchers, such as Yang et al., developed their tools to enable gbXML reconstruction and improve the geometric interoperability between BIM and BEM, focusing mainly on geometry-related aspects [21]. Finally, other studies, like that of Kamel and Kazemian, integrated their automation within software already used by professionals to conduct BIM-integrated thermal analysis [54]. Although resolving interoperability issues, these approaches focus on optimizing the conversion process from BIM to BEM rather than on the joint generation of the two in a manner already optimized for interoperability, which is the goal of this study.

3. Methods and tools

3.1. Theoretical framework

The theoretical framework for defining TBIM is based on the following theoretical principles, defined as 'spatial reasoning' (subsection 3.1.1), 'conditional information modeling' (subsection 3.1.2), 'semi-automation' (subsection 3.1.3), 'semantic flexibility' (subsection 3.1.4), and 'progressive data enrichment' (subsection 3.1.5).

3.1.1. Spatial reasoning

Spatial reasoning in BIM involves structuring building information around spatial objects instead of physical elements. This perspective prefers using geometrically succinct and conceptual digital models. It contrasts the current course of many BIM processes that develop highly detailed digital models to precisely depict projects' three-dimensional form and inventory [46], an approach that can create gaps in the semantic content of BIM models and, at the same time, lead to information overproduction when details are not needed [55].

The BIM approach presented in this research uses spatial reasoning by aligning its vision with Topologic [45], relying on the core idea of thinking of buildings as assemblages of topologically connected spaces (or, more generally, spatial entities) capable of hosting occupants while satisfying various needs, encompassing formal, functional and performance-related aspects. From this perspective, the 'space' entity plays a central role in the architectural representation process. The way we use spaces strongly influences the daily life of buildings. Their use is directly linked with the occupancy conditions, the activities conducted within them, and, therefore, the ways people live in architecture. Spaces become central to building management since they are the primary environments serving the areas where people reside and work, housing the systems required to create favourable living conditions. For these reasons, this study considers spaces the ideal collectors of information concerning buildings' management aspects.

3.1.2. Conditional information modeling

'Conditional information modeling' (or 'rule-based information modeling'), usually associated with model validation and checking activities [56,57], refers to the process of modeling building knowledge by semi-automatically assigning data to spatial and construction elements on the basis of predefined topological and semantic rules and conditions. These conditions can pertain to factors such as element location and adjacencies, element types, thermal requirements, acoustic specifications, or any other functional, geometrical or performance-related criteria.

The method we present fosters conditional information modeling to enhance automation and control within the information modeling process by adopting Topologic's class hierarchy. This is a hierarchical structure founded on multiple classes capable of representing building topology using abstracted and interconnected geometries (i.e., 'Cluster', 'CellComplex', 'Cell', 'Shell', 'Face', 'Wire', 'Edge', and 'Vertex') that allows efficient rule-based query operation in modeling and retrieving information. As will be illustrated in detail in the next section, the TBIM approach assumes that a building can be represented as a Topologic cell complex, a space as a Topologic cell, a partition element as a Topologic face, and an opening as a Topologic aperture. Adhering to this spatial hierarchy, rule-based information assignments can be done by executing topological queries. To make a practical example, through conditional modeling, a specific property value (e.g., U-Value) can be assigned to all the internal vertical partitions within a specific building that separate a heated space from an unheated space to ensure the satisfaction of specific energy performance requirements. Similarly, another property value may be assigned to the vertical partitions separating a heated space from another, and so on.

3.1.3. Semi-automation

The conditional modeling approach thus allows for semi-automated semantic data enrichment in the BIM process. This procedure is denoted as 'semi-automated' as manual-made rules and conditions are implemented and used to determine how properties are assigned to different elements within the BIM model. Unlike full automation, semi-automation involves human operators in rule assignment, thereby preserving human agency in modeling the building knowledge. For heritage building representation, semi-automation provides human control over information and information flows, proving useful for critically interpreting the building's composition.

3.1.4. Semantic flexibility

Semantic flexibility is crucial for allowing BIM to be utilized with external third-party applications, such as performance simulations. Current BIM software often struggles to interpret information that is not explicitly defined in either native or universally recognized BIM schemas, like the IFC [40]. For example, standard BIM cannot currently capture and store dynamic information, such as the data gathered by sensor systems or results generated from dynamic performance analyses [58,59]. In the realm of performance-based design and management, achieving semantic flexibility is essential for ensuring that various digital platforms and models can work together seamlessly. This interoperability is critical to applying specialized knowledge from various domains to uses that extend beyond traditional BIM tasks.

Our method employs a graph-based approach for modeling building entities to enhance semantic flexibility. Specifically, building representations are constructed using Topologic's 'Graph' class. Graph data structures are versatile tools for storing various data types, including objects, relationships, and attributes. This format is especially well-suited for preserving diverse data types. Furthermore, graphs facilitate efficient information retrieval through semantic queries [60]. In this research, graphs serve as intermediate repositories for storing and accessing enriched BIM data. They also enable semantic connections with data from other sources, such as BEM.

3.1.5. Progressive model enrichment

The last principle guiding the research is progressive data enrichment. When digitally modeling existing buildings, progressive data enrichment means acquiring and assigning difficult-to-find data only when available and effectively needed. For instance, in the context of building management, a simple initial model may only contain basic information about space dimensions. As the model is well-conceptualized, the spaces can be gradually enriched with new information (such as energy requirements for conducting energy audits or safety requirements for planning safe occupancy) as necessary. The knowledge process is, therefore, dynamic and iterative, and it can evolve throughout the building's lifecycle in response to emerging needs over time. This adaptability, combined with a good understanding of the informational and relational implications, a well-structured ontology, and a straightforward embedded knowledge structure, can empower the construction of digital models with remarkable capabilities.

The proposed methodology achieves progressive model enrichment using Topologic's 'Dictionary' class. As anticipated, in Topologic, object classes like vertices, edges, wires, and faces act as geometric abstractions for more detailed representations. By employing custom dictionaries, these objects can store essential information, including attributes necessary for simulation or future detailing work. This iterative process allows the digital model to grow over time, accommodating the data needs as they evolve.

3.2. Methodological workflow

This section illustrates the methodology developed for delivering the TBIM models and deriving the BEMs from them. The application of the methodology is grounded on a specific spatial hierarchy (subsection 3.2.1) and comprises five conceptual steps (subsection 3.2.2), which are: (1) 3D modeling, (2) topology modeling, (3) information enrichment, (4) BIM modeling, and (5) BEM modeling.

Fig. 1 depicts the structure of the workflow and the toolchain adopted. The workflow can vary based on the available inputs, adapting to different input entries and tools. Topologic is used as a modeling environment in the workflow to generate the TBIM. Autodesk Revit is instead chosen as the BIM modeling environment. Python serves as the programming language for developing the functions to create the TBIM models. Specifically, the Topologicpy package [23] is the core of these functions. Since Topologicpy does not have a graphical user interface (GUI), PyRevit [61] allows the user to apply the modeling steps within Autodesk Revit, acting as its plugin. Finally, Ladybug Tools [24], in particular Honeybee (HB), are used to realize the BEM and launch energy analysis.

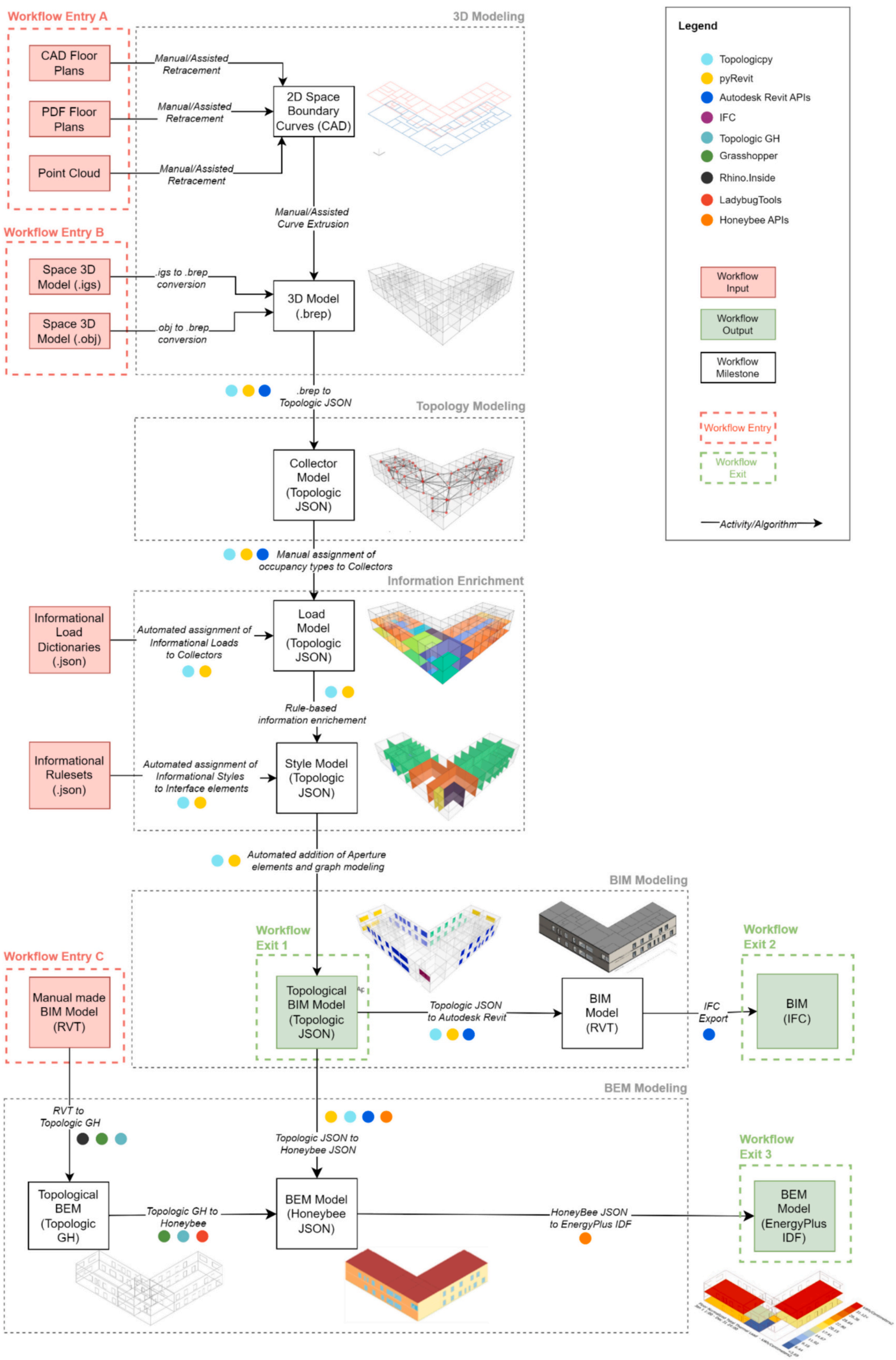


Fig. 1. Workflow for generating TBIM, BIM and BEM models.

3.2.1. Spatial hierarchy

The basic assumption is that, in the workflow, information is modeled according to a predefined spatial hierarchy independently from the tool used. This spatial hierarchy has been defined by aligning the definitions and concepts of some relevant data schemas in the AECO industry, namely IFC for BIM, EnergyPlus (EP) Input Data Format (IDF) for BEM, and Topologic for topology modeling. IFC, introduced in Section 2, represents the international open specification for BIM data. [40]. EP is an open-source BPS calculation engine program developed by the US Department of Energy, used worldwide for evaluating buildings' energy performance, environmental impact, and thermal comfort [25]. Its use is so widespread that its schema, the IDF format, can be considered almost a standard for energy modeling. Topologic, mentioned in the previous section, is an open-source software library developed by Cardiff University in collaboration with University College London (UK) to model building topology [45].

The spatial hierarchy, visually illustrated in Fig. 2, stands that:

- A building can be represented as a IfcBuilding, an IDF Building, or a Topologic CellComplex.
- A space within the building can be represented as an IfcSpace or a Topologic Cell.
- A zone within the building, intended as a group of spaces, can be represented as an IfcZone, IDF Zone, or Topologic Cluster.
- Partition elements, including walls, roofs, and slabs, delimiting the spaces within the building can be represented as IfcWalls, IfcRoofs, IfcSlabs, IDF BuildingSurfaces, or Topologic Faces.
- Openings, including windows, doors, and holes, hosted in a partition element can be represented as IfcWindows, IfcDoors and IfcOpenings, IDF FenestrationSurfaces, or Topologic Apertures.

This spatial hierarchy can be also aligned with BOT and Brick schemas for LBD applications. Specifically, a building can be represented by the 'Building' class in both BOT and Brick. A zone can be defined as a 'Zone' in both schemas. A space can be considered a space 'Zone' in BOT and a 'Space' in Brick. Partition elements can be described as 'Interfaces' in BOT; however, they do not have a counterpart in Brick. BOT generally treats Openings as 'Elements', while Brick does not explicitly address them.

3.2.2. Workflow structure

The five methodological steps introduced before are detailed below.

3.2.2.1. Step 1: 3D modeling. The first step involves creating the geometry of the building. This is achieved by modeling a closed 3D volume for each space within the building as a BRep object. This object, which represents the gross geometry of the space, is then converted into a Topologic cell, the basic spatial element within the building's model. Depending on available tools, the 3D model can be made manually or through automated processes. For instance, as in this paper, CAD or PDF drawings can be retraced to extract, manually or automatically, the gross boundary curves of the spaces and subsequently extrude them into a three-dimensional format. Similarly, point clouds can be segmented and processed manually or automatically to derive the profiles delimiting the spaces and extruding them to create closed 3D volumes.

3.2.2.2. Step 2: Topology modeling. In the second step, the topological relationships between the essential elements of the model are created. At this stage, although the cells do not have any information attached, they are ready to be filled with new data. For this reason, they are called 'Informational Collectors', as they serve as the main data collectors in the modeling process.

Specifically, to transform the geometrical elements into topological elements, the Topologic cells are aggregated into a higher-order spatial entity, i.e. the cell complex. This operation, conducted thanks to Topologicpy's 'Topology. ByBRrep' method within PyRevit, allows the linking of each cell composing the building to each other cell through face adjacency relationships. The outcome of this step is the 'Collector Model', a Topologic cell complex in the Topologic JSON format composed of interconnected cells and faces.

3.2.2.3. Step 3: Information enrichement. In the third phase, information is assigned to the elements composing the cell complex (i.e., the cells and the faces). This procedure is performed in PyRevit through conditional modeling with the primary objective of semi-automatically setting the data needed for energy analysis.

First, functional data is added to the informational collectors by attaching the so-called 'Informational Load Dictionaries' (ILD). These consist of JSON dictionaries, each corresponding to a specific space function (e.g., office, classroom, corridor, restroom, etc.) and containing related data (e.g., temperature, humidity, ventilation and lighting setpoint values, as well as occupancy density and people capacity, but not only). To enrich the collectors with new information, a specific space function is assigned to each collector and the corresponding ILD is transferred to the respective cell, enriching it with the data related to the chosen function. For example, a certain temperature setpoint value (e.g., 20 °C) can be assigned to all the offices by defining it in an ILD designed explicitly for office spaces. Similarly, another setpoint value can be set for all the corridors (e.g., 16 °C).

Second, after adding the data to the collector cells, this data **is** also

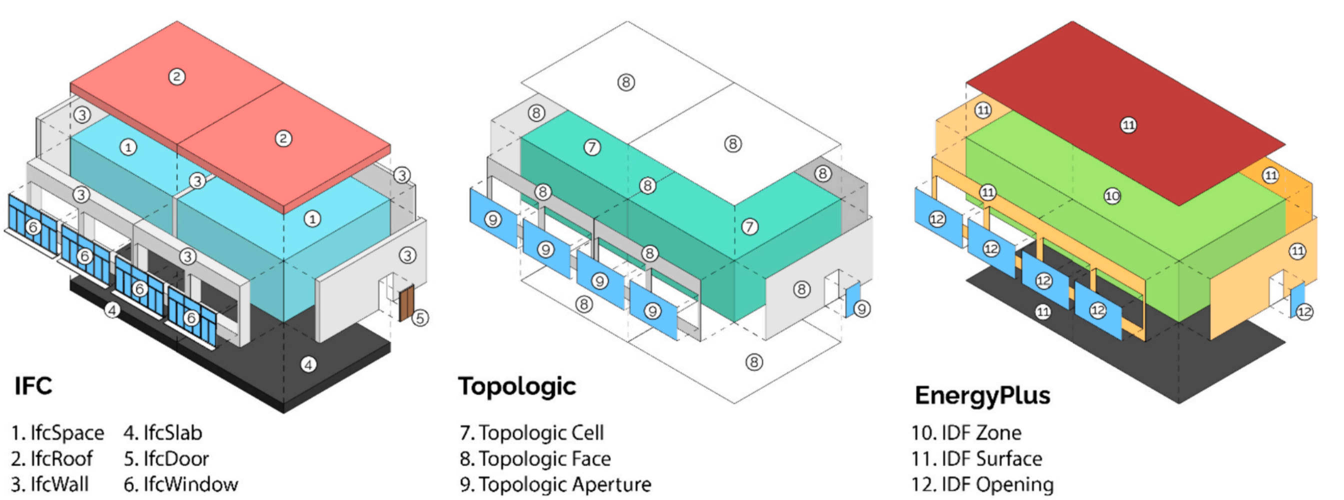


Fig. 2. Alignment of IFC, IDF, and Topologic schemas according to the spatial hierarchy.

transferred to the adjacent faces by executing topological queries. The faces belonging to the cell complex are classified according to their topological type as 'internal vertical', 'external vertical', 'internal horizontal', 'bottom horizontal', and 'top horizontal'. Then, the query is executed, and data is attached to the faces as a Topologic dictionary. This procedure is iterated for each face by (a) querying the cells adjacent to the face, (b) extracting the information attached to these cells, (c) creating a new Topologic dictionary containing the extracted information, (d) and transferring the new dictionary to the face. Following the previous example, through this process, an internal vertical face adjacent to an office on one side and a corridor on the other can be designated as separating a space at 20 °C from a space at 16 °C, along with other properties. The result of this step is called the 'Load Model', a Topologic cell complex containing the ILDs' information.

Subsequently, the faces undergo further data enrichment. This enrichment is achieved using the so-called 'Informational Rulesets' (IRSs). An IRS is a data dictionary collecting 'conditions' and 'styles' applicable to the faces. The conditions dictate the property values a face should have so that the IRS can be applied to the face itself, while the styles represent the new data to be assigned to the face if it meets the specified conditions. In simpler terms, when all the conditions of an IRS match the properties of a face, the styles' data is attached to that face. The assignment of styles' data also occurs through topological queries. All the IRS dictionaries are iterated over each face within the cell

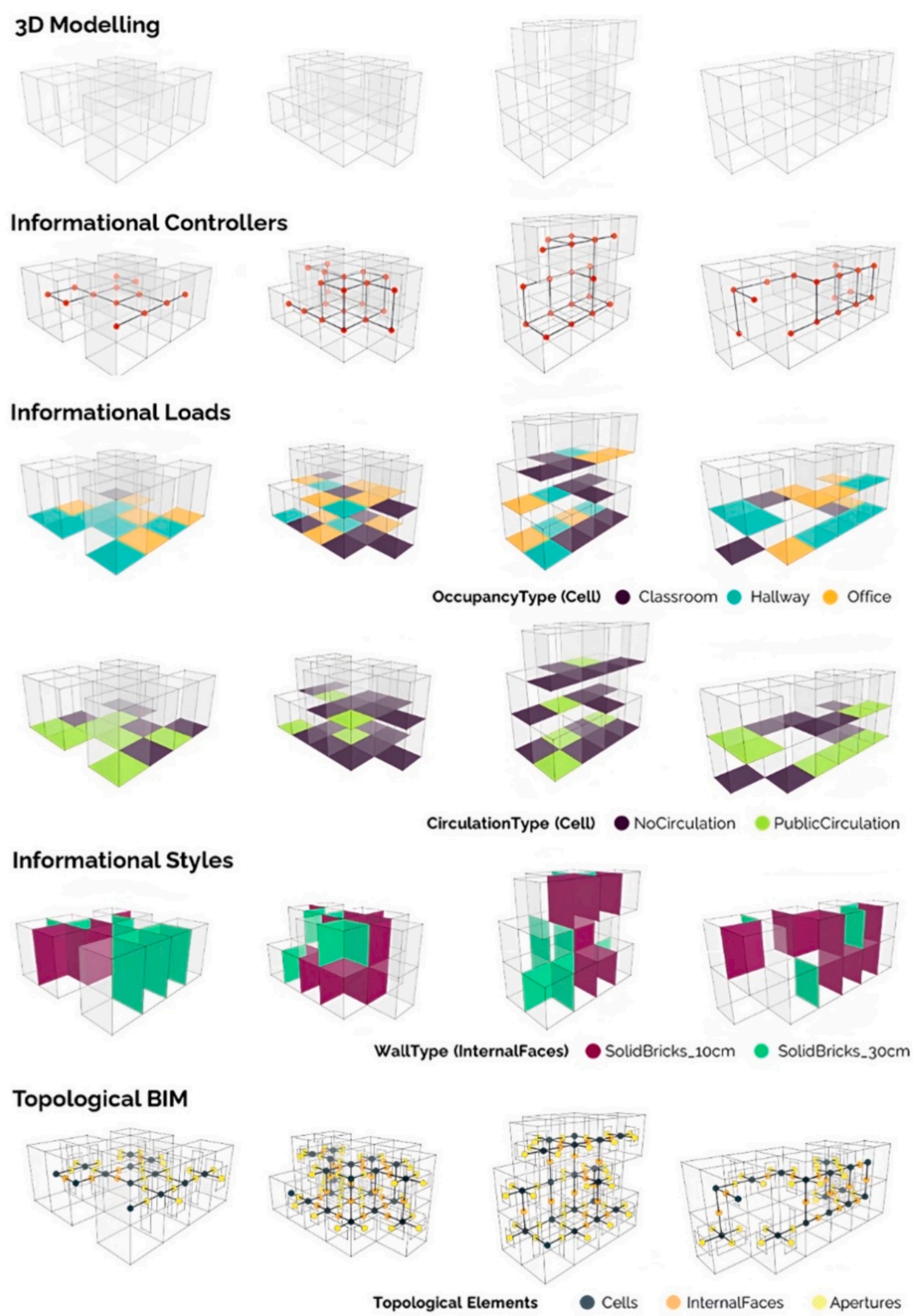


Fig. 3. Workflow's conceptual demonstration. From the top to the bottom: (a) 3D modeling of some example buildings in Topologicpy; (b) creation of informational collectors; (c) assignment of informational loads; (d) assignment of styles to internal faces; e) generation of TBIM models with added openings and graph visualization.

complex. For each face, the conditions' values are accessed and compared to the face's properties. If the values match, a new dictionary containing the styles' properties (and data) is created and added to the face; otherwise, the iteration continues. In the case of multiple IRSs matching with the same face, styles' values are overwritten. The outcome is the so-called 'Style Model', a Topologic cell complex in the Topologic JSON format that contains both the ILDs' and IRSs' data.

In this study, this conditional data enrichment process is applied to the Load Model to assign construction and aperture data to the faces. For instance, a specific U-value can be set for all external vertical faces adjacent to heated or unheated spaces. Or, a certain number and type of doors or windows can be assigned to all the faces adjacent to the cells with a certain function, and so on.

3.2.2.4. Step 4: BIM modeling. At the beginning of the fourth stage, the cells and the faces composing the cell complex are already informed with all the data assigned through the procedures described in the previous passages, which mainly include indications about the functional and energy requirements of the spaces and the construction characteristics of the faces. This data is used within PyRevit to generate the TBIM and BIM models at this step.

To complete the modeling procedure, the apertures of the building are first created. These consist of doors, holes, and windows; doors provide horizontal passage between horizontally adjacent cells, holes between vertically adjacent cells, and windows between the cells and the external environment. The apertures are created as face elements in Topologicpy on the basis of the data attached to the faces, which include information about the size of the apertures, their material type and thermal properties. Once the geometries of the apertures are created, they are linked to the related data through new Topologic dictionaries and added to the Style Model thanks to the 'Topology.AddApertures' method of Topologicpy. Then, the cell complex is transformed into a Topologic graph and graph visualization and analysis are used to check if the modeling procedure produced errors and, in this case, to correct them. The result is the 'Topological BIM Model', a conceptual model consisting of a Topologic cell complex composed of cells, faces, and apertures semi-automatically informed with data useful for energy analysis. This model is not only a simple collection of spatial and topological elements but a system of objects interrelated through topological relationships suited for a direct transformation into BIM and BEM models. For visual clarity, Fig. 3 depicts the application of the first four stages of the workflow in some simple examples of buildings built in Topologicpy.

Starting from the Topologic TBIM, an Autodesk Revit BIM model is then automatically created for export to IFC. Technically, this procedure involves using Topologicpy and Autodesk Revit APIs within PyRevit to convert the Topologic cell complex into a Revit building. This conversion is performed by aligning Topologic's class hierarchy with Revit's element classes and using Autodesk Revit API methods to convert Topologic elements into Revit elements. Specifically, Topologic cells are converted into Revit spaces, Topologic's vertical faces into Revit walls, Topologic's horizontal faces into Revit floors and roofs, and Topologic's apertures into Revit's windows, doors and holes. The technical procedure for applying the conversion is further detailed in Section 4.

3.2.2.5. Step 5: BEM modeling. In the final step, a BEM model is derived from the previously generated TBIM. This operation is carried out using an approach similar to the Topologic-to-Revit conversion but aligning the element classes and properties of the Topologic TBIM with those of Ladybug Tools, specifically HB, through the Ladybug Tools APIs in Python. To apply the model translation, the TBIM's cells are first aggregated into Topologic clusters, representing the thermal zones of the buildings, and cells' data are transferred to the clusters by executing data aggregation operations. Then, the faces of the TBIM are converted into HB wall, roof, and floor surfaces. Similarly, the Topologic apertures are converted into HB window surfaces. An HB zone is created for each Topologic cluster, and the energy-related data are transferred. The outcome is a BEM model in the HB JSON format, then converted into an IDF model (thanks to the 'honeybee.model' module of the HB APIs) that can be inputted into EnergyPlus to conduct energy simulations.

In cases where the building's BIM model already exists in Autodesk Revit, and there is no need to create a TBIM from scratch using the steps previously described, the BEM model can still be topologically created by following a different procedure, which was already outlined in authors' previous research [62]. This process involves creating an intermediate topological energy model (TBEM) to facilitate data transfer between BIM and BEM. In this workflow, BIM, TBEM and BEM are parallel and interconnected models, each serving a specific function and storing the data needed to perform that function in the most adequate environment. The BIM furnishes the semantic and information framework; the TBEM abstracts the geometry from the BIM and prepares it for the BEM, selects the model view definition required for simulation purposes, and transfers energy-related data; the BEM stores all these data and uses them to perform the energy analysis. In this case, the BIM to BEM conversion is achieved through VP scripts utilizing BIM and BEM APIs, allowing direct interactions with BIM and BEM systems. Specifically, Grasshopper serves as the VP platform with the assistance of the Rhino.Inside.Revit plugin [63] for integration with Autodesk Revit. The TBEM is constructed using Topologic within GH, while the Ladybug Tools within Grasshopper are employed to configure the BEM and execute simulations via the EP engine.

4. Technical implementation

4.1. Case study

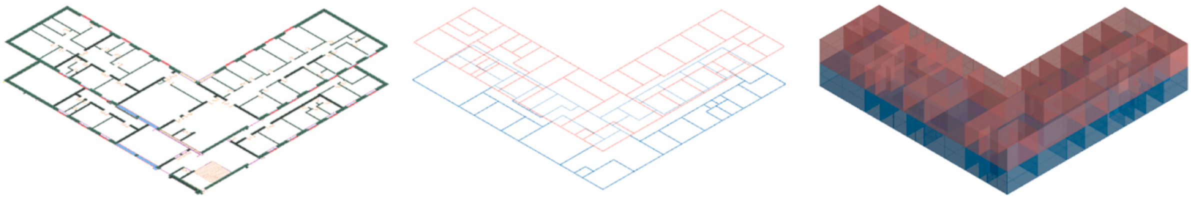
The Faculty of Engineering in Bologna (Italy) [64] is considered as a pilot case study for the workflow. Built between 1932 and 1935, it is one of the first 20th-century buildings listed in the city. With its 19,200 sqm of net floor area and four levels, its maximum capacity amounts to 5000 users (including researchers, employees, and students), with approximately 2500 students using the building according to the academic timetable hours 5 days a week, 11 months a year. In particular, a part of the building is considered for demonstration. This part houses the Department of Architecture (DA), identified as 'BlockA', and the Department of Civil, Chemical, Environmental, and Material Engineering (DICAM), identified as 'BlockB'. This building area is particularly significant as it encompasses various functions representing the functional complexity that usually characterizes higher-education buildings, like offices, meeting rooms, libraries, circulation spaces, and classrooms.

4.2. 3D modeling

The 3D modeling of the case study was conducted using Rhino software [65]. The first 3D modeling operation involved reproducing the gross curves demarcating the spaces by manually retracing the building's CAD floor plans (Fig. 4). The space curves were retraced as closed polylines lying on XY planes with elevations equal to that of the building storeys. After retracing, a GH algorithm was used to optimize the boundary curves and ensure that the edges and vertices of adjacent curves matched, which is a crucial prerequisite both for modeling building topology in Topologic and running energy analysis without errors. Following this, the curves were extruded upwards to their top floor level and, consequently, the extruded poly-surfaces were capped to obtain closed volumes, then exported as BReps thanks to the 'Topology.ExportToBRep' node of TopologicGH.

4.3. Topology modeling

The Topologic cell complex was generated to model the topology of the case study (Fig. 5). This was performed within PyRevit through the



(a) (b) (c)

Fig. 4. 3D modeling of the case study: (a) CAD drawings, (b) retraced gross space boundary curves, (c) closed space BReps.

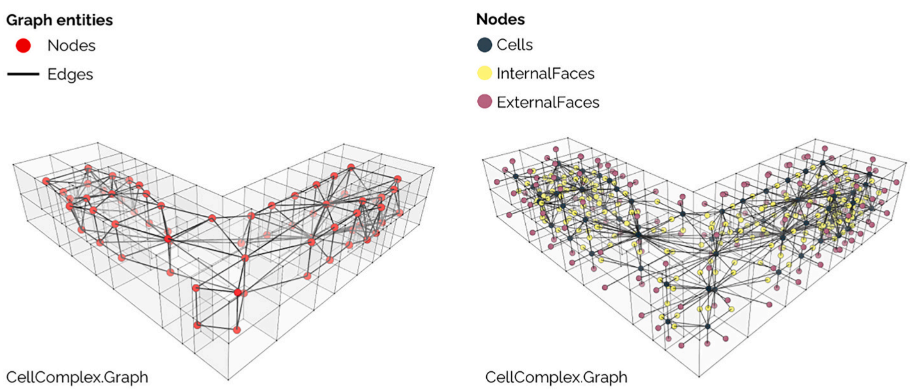


Fig. 5. Collector Model in Topologicpy. Graph visualization highlights adjacency relationships between cells (a) and between cells and faces (b).

'Topology.ByBREPPath' function of Topologicpy by taking the .brep file previously created as input. After that, the cells of the Collector Model were converted to simple Revit spaces thanks to PyRevit and Autodesk Revit APIs, enabling the user to interact with the Informational Collectors through Revit's user interface. This operation provided two outcomes, shown in Fig. 6: (a) the Collector Model, i.e. the Topologic cell complex, saved as a JSON file, containing only topological information about all the cells belonging to the cell complex, and (b) the informational collectors corresponding to the cells of the Collector Model in Revit. These collectors were treated as spaces inside Revit to enable the modeler to use Revit's GUI to interact with the Topologic model (running in the background) during the information enrichment process.

```
{
 'pr_OccupancyType': 'Office:ResearchOffice',
 'pr_CirculationType': 'NoCirculation',
 'pr_AreaPerOccupantPeak': 8.0,
 'pr_IsOccupied': 1,
 'pr_IsHeated': 1,
 'pr_IsCooled': 1,
 'pr_IsNaturallyVentilated': 1,
 'pr_IsMechanicallyVentilated': 0,
 'pr_SpaceTemperatureWinterMax': 20,
 'pr_SpaceTemperatureSummerMin': 26
 'pr_SpaceTemperatureSummerMax': 30,
 'pr_SpaceTemperatureWinterMin': 16,
 'pr_HumidityMin': 0,
 'pr_HumidityMax': 100,
 'pr_Illuminance': 300,
 'pr_NaturalVentilationRate': 7.0,
 'pr_EquipmentPowerDensity': 5.0,
 'pr_LightingPowerDensity': 10.0,
 'pr_MechanicalVentilationRate': 0.0,
}
```

Fig. 6. Example of ILD in JSON format. The ILD contains information about occupancy and energy requirements of research office spaces.

4.4. Information enrichment

4.4.1. Creation of the informational load dictionaries

The ILDs were created in a JSON file to enrich the model with information and assigned to the collectors. As already explained, these consist of JSON dictionaries, each corresponding to a specific type of space occupancy and storing related information, which enrich collectors with information about functional and energy characteristics. An example of ILD referring to research office spaces is provided in Fig. 6. This ILD stores data about the energy and occupancy requirements of research offices. This information includes data such as the type of circulation, the area per occupant during peak hours, indications of whether the space is occupied, heated, cooled, and ventilated, as well as temperature, humidity, lighting setpoints, ventilation rates, and the electrical load for appliances and lights.

In the case of applications of the TBIM that are different from energy analysis, the ILDs can be personalized to include other performance-related data, such as fire safety and acoustic requirements.

4.4.2. Assignment of ILDs

Once the ILDs were created, they were assigned to the informational collectors by interacting with Revit's interface through PyRevit. This operation was carried out through a dedicated function that allowed for manually selecting the Revit spaces, assigning the occupancy types to them, and automatically uploading the related information to the model's cells (Fig. 7). Together with occupancy types, all the information incorporated in the ILDs was transferred to the collectors (Fig. 8). This process was achieved through a Python function that first read the occupancy types of the Topologic cells and then, for each cell, transferred the properties of the corresponding ILD both to the Revit spaces (by creating new properties in a shared parameter file and assigning the corresponding values to the spaces), and to the Topologic cells (enriching the cell's dictionaries new key-value pairs). The correspondence between the Revit spaces and the Topologic cells was achieved by using the exact unique identification of the collectors in the two different BIM environments.

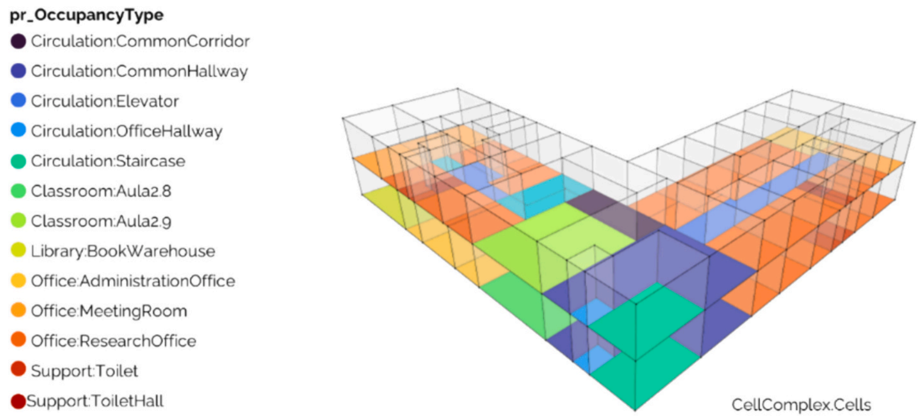


Fig. 7. Cell complex after the assignment of Informational Loads. Cells are coloured by occupancy type.

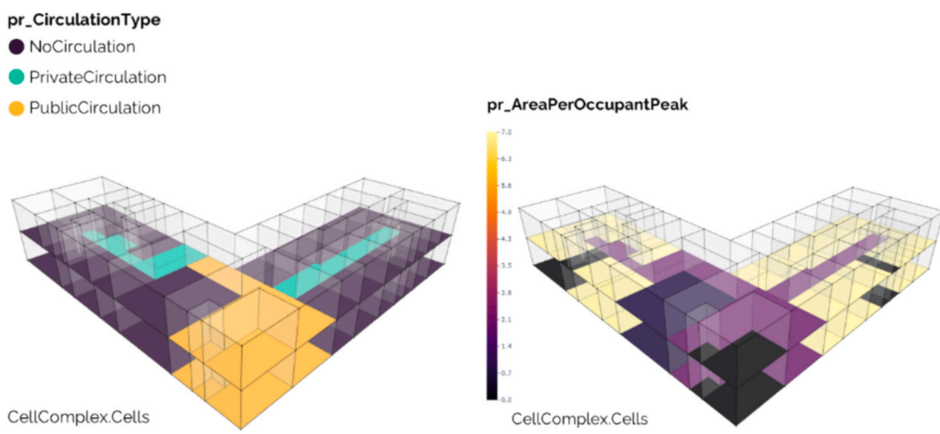


Fig. 8. Cell complex after the assignment of Informational Loads. Cells are coloured by circulation type and area per occupant at peak.

Consequently, the ILDs' data was transferred to the faces composing the cell complex by executing adjacency topological queries. This procedure was also carried out using some Python functions developed in PyRevit. Through the function, for each face of the Topologic model, the adjacent cells were identified using the 'Topology.AdjacentTopologies' method of Topologicpy. Then, the values of the cells' dictionary properties were read and transferred to the face. For example, an internal vertical face adjacent to a cell with 'pr_CirculationType' equal to 'NoCirculation' and another cell with 'pr_CirculationType' equal to 'PrivateCirculation' was enriched with the 'pr_CirculationType' property equal to 'NoCirculation-PrivateCirculation', and so on for the other properties (Fig. 9).

Data regarding the exposure of the vertical faces was also added in this phase. This information was obtained by setting the project north in Revit and performing simple geometric operations on face normals through Topologicpy within PyRevit. The outcome of this step was the 'Load Model', a cell complex with the cells and faces informed of the ILD data.

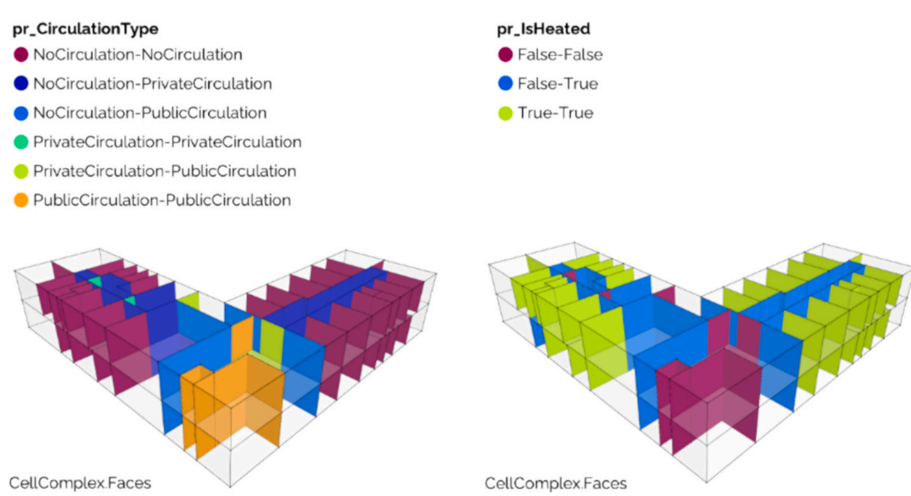


Fig. 9. Internal vertical faces informed from adjacent cells.

4.4.3. Creation of the informational rulesets

Next, the IRSs were created and used to assign the informational styles to the faces of the cell complex. The IRSs were created in PyRevit using specialized functions that offered a GUI for generating the rulesets as JSON dictionaries. These functions allowed to produce a JSON file compiling all the IRSs, formatted as in the example of Fig. 10, to ensure compatibility with the algorithms developed for applying conditional data enrichment in PyRevit.As mentioned in Section 3, the IRSs are composed of conditions and styles. Conditions express the rules that the faces must satisfy. Styles instead represent the new properties to assign to the faces when the rules are respected. Specifically, the style properties in Table 1 were used to model the case study's energy-related attributes.

4.4.4. Assignment of the informational rulesets

The IRSs, stored as a list in a dedicated JSON file, were applied iteratively to each face of the Topologic cell complex. In the assignment procedure, the condition properties were first compared to those of the face for each IRS. Then, if all the conditions matched the properties of the face, the style properties were transferred to the face. If multiple IRSs matched the conditions of the same face, the style property of the last IRS was overwritten in the face's dictionary. This approach, therefore, required modeling the IRSs starting from the most general (e.g., assign a certain wall type to all the vertical faces) to the more specific (e.g., assign a certain wall type properties to those faces that are adjacent to a cell with certain occupancy type and a certain thermal requirement).

The result of this step was the 'Style Model', a cell complex with the faces informed of the IRSs' data (Fig. 11 and Fig. 12).

4.5. BIM modeling

Starting from the Style Model, i.e. the Topologic cell complex enriched with the informational styles, the TBIM model was generated first in Topologic, and then the BIM was derived from it in Autodesk Revit.

To generate the Topologic TBIM, first, the style properties were read for each face of the cell complex and based on them, the apertures hosted within the faces were added to the Topologic model. More specifically, a PyRevit function was developed to determine the number of door, hole, and window apertures by reading the 'WindowCount,' 'DoorCount,' and 'HoleCount' values and retrieving the aperture sizes from the faces' dictionaries (e.g., 'WindowHeight,' and 'WindowWidth'). Based on these sizes, the apertures were created using the Topologic's 'Face. Rectangle' method and then added to the cell complex, and the style data (e.g., 'UValue', 'SolarHeatGainTransmittance' and 'VisualTransmittance') were transferred to them. By default, this function models the apertures with a rectangular shape and positions them equidistant within the face, so they do not overlap. Using this technique, holes were also created in the internal horizontal faces adjacent to staircases and elevators. The result was the Topological BIM (Fig. 13), a Topologic cell complex with fully interconnected cells, faces and apertures enriched with functional and energy-related data about the spaces and material specifications necessary for energy simulation.

The TBIM was then converted into a Revit model by aligning Topologic's with Revit's classes and using the Autodesk Revit APIs methods [66] for converting Topologic's elements into Revit's ones (Table 2).

First, the elevation values of all horizontal faces in Topologic were read using Topologicpy's functions in PyRevit, and for each elevation found, Revit levels were created using the 'Level.Create' command from Revit's APIs. Then, the Topologic vertical faces were converted into Revit walls by extracting their planar geometric profile through Topologicpy and assigning it to the 'Wall.Create' by profile method from the Revit APIs. In this step, the wall types shown in Fig. 11 (previously modeled in Revit with names corresponding to those in the Topologic models) and all the material characteristics necessary for energy analysis were assigned to the Revit walls. Based on their geometric characteristics, all walls were associated with a lower and upper level in Revit. Furthermore, all the properties contained within the Topologic faces' dictionaries were transferred to the Revit walls by creating new shared parameters (named as the keys in the Topologic dictionaries) and setting the corresponding values. Similarly, the Topologic horizontal faces were converted into Revit floors (using the 'Floor.Create' method) if they

```
{
  'cnFC_equalTo_pr_TopologicalType': 'ExtVer',
  'cnFC_equalTo_rl_HasLocation_tpBB': 'BlockA',
  'cnFC_contains_pr_OccupancyType': 'Office',
  'cnFC_equalTo_pr_FaceExposure': 'W',
  'stFC_WallType_Name': 'WA_Ext_BrickWall_79-(2+1)',
  'stAP_WindowCount': 1,
  'stAP_WindowHeight': 275,
  'stAP_WindowWidth': 200,
}
```

Fig. 10. Example of IRS in JSON format. This IRS applies the wall type and the size of windows to external vertical faces located in Block A, exposed to West, and adjacent to office spaces.

Table 1
Style properties for modeling construction and opening data.

Property Name	Description
FloorTypeName	It specifies the name of the Revit floor type to assign to the face to detail its material properties in the BEM. The element type includes information about the floor's material layers, including thickness, thermal conductivity, density, specific heat, and thermal transmittance.
RoofTypeName	It specifies the name of the Revit roof type to assign to the face to detail its material properties in the BEM, as for the floors.
WallTypeName	It specifies the name of the Revit wall type to assign to the face to detail its material properties in the BEM.
ApertureCount	It specifies the number of apertures hosted by the face, including both windows and doors, to be added to the cell complex.
DoorCount	It specifies the number of doors hosted by the face to be added to the cell complex.
HoleCount	It specifies the number of holes hosted by the face to be added to the cell complex.
DoorHeight	It specifies the height of the doors hosted by the face in centimetres.
DoorWidth	It specifies the width of the doors hosted by the face in centimetres.
DoorShutterNumber	It specifies the number of shutters hosted by the face. It can be either 1 or 2.
WindowHeight	It specifies the height of the windows hosted by the face in centimetres.
WindowWidth	It specifies the width of the windows hosted by the face in centimetres.
WindowsSillHeight	It specifies the sill height of the windows hosted by the face in centimetres.
UValue	It specifies the thermal transmittance of the apertures hosted by the face in W/m^2K.
WindowFrameMaterial	It specifies the material of the window frames hosted by the face as a string (e.g., 'wood').
WindowFrameUValue	It specifies the thermal transmittance of the window frames hosted by the face in W/m^2K.
WindowGlassLayerCount	It specifies the number of glass layers the windows hosted by the face as an integer.
WindowGlassUValue	It specifies the thermal transmittance of the glass in the windows hosted by the face in W/m^2K.
WindowSolarHeatGainTransmittance	It specifies the solar heat gain transmittance of the glass in the windows hosted by the face as a float (ranging from 0 to 1).
WindowVisualTransmittance	It specifies the visual transmittance of the glass in the windows hosted by the face as a float (ranging from 0 to 1).

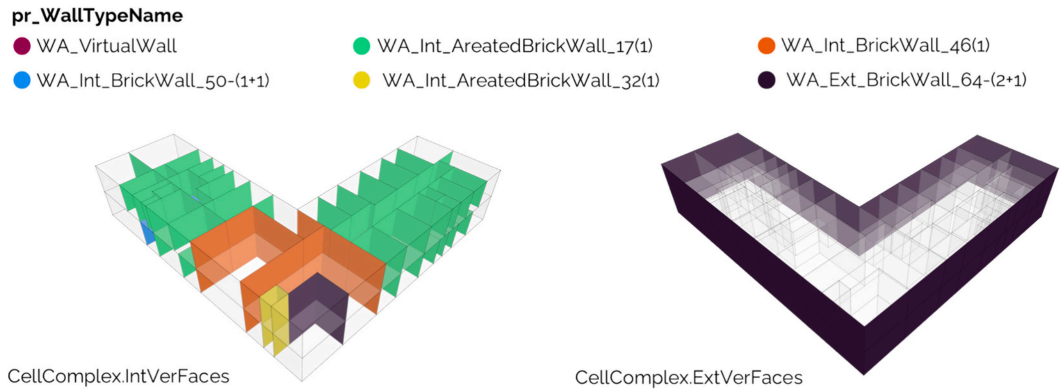


Fig. 11. Wall types set for vertical faces.

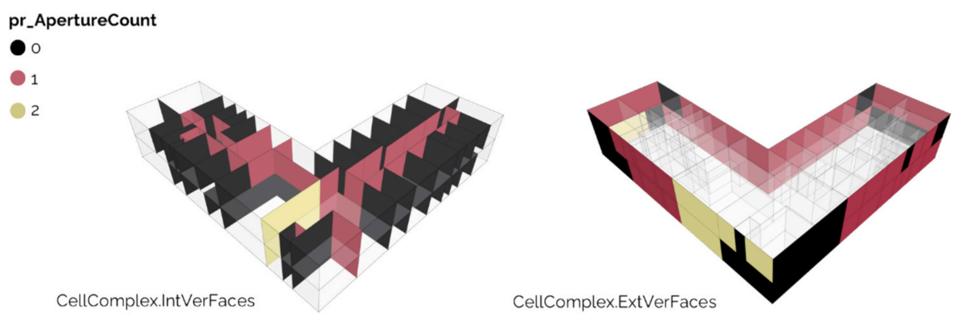


Fig. 12. Number of apertures hosted by vertical faces.

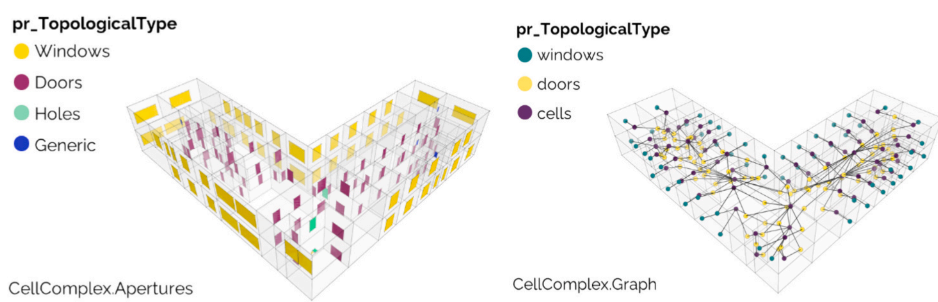


Fig. 13. TBIM in Topologicpy. On the left, apertures highlighted by topological type. On the right, graph visualization.

Table 2
Autodesk Revit APIs' methods for converting Topologic's elements to Revit's ones.

Topologic class	Revit class	IFC class	Revit API Method
–	Level	IfcBuildingStorey	Level.Create
Face (external and internal vertical)	Wall	IfcWall	Wall.Create
Face (bottom and internal horizontal)	Floor	IfcSlab	Floor.Create
Face (top horizontal)	Roof	IfcRoof	Create.NewFootPrintRoof
Cell	Space	IfcSpace	Create.NewSpace
Aperture (door, hole)	Door	IfcDoor	Create.NewFamilyInstance
Aperture (window)	Window	IfcWindow	Create.NewFamilyInstance

were in contact with the ground or internal, and into Revit roofs (using the 'Create.NewFootPrintRoof' method) if they were on the top.

Once all faces were modeled, the apertures were added to the Revit model by transforming the Topologic apertures into Revit doors and windows. A placeholder family of doors and windows was previously modeled in Revit to model apertures, and the width and height values of the Topologic apertures were read. New family types were created in Revit based on height and width properties. Then, all properties of the Topologic apertures were transferred from Topologic to Revit, as done for the faces.

Finally, the Topologic cells were transformed into Revit Spaces. The center of mass of the Topologic cells was found thanks to Topologicpy and assigned as the geometric point for creating the Revit space using the 'Create.NewSpace' method of Revit APIs. Depending on the position of the center of mass, the space level and the level that bound it from above were automatically calculated. As with faces and apertures, all properties from the Topologic cells' dictionaries were transferred to the Revit spaces. The output was a Revit BIM model containing all the information elaborated throughout the workflow (Fig. 14), which was exported to IFC.

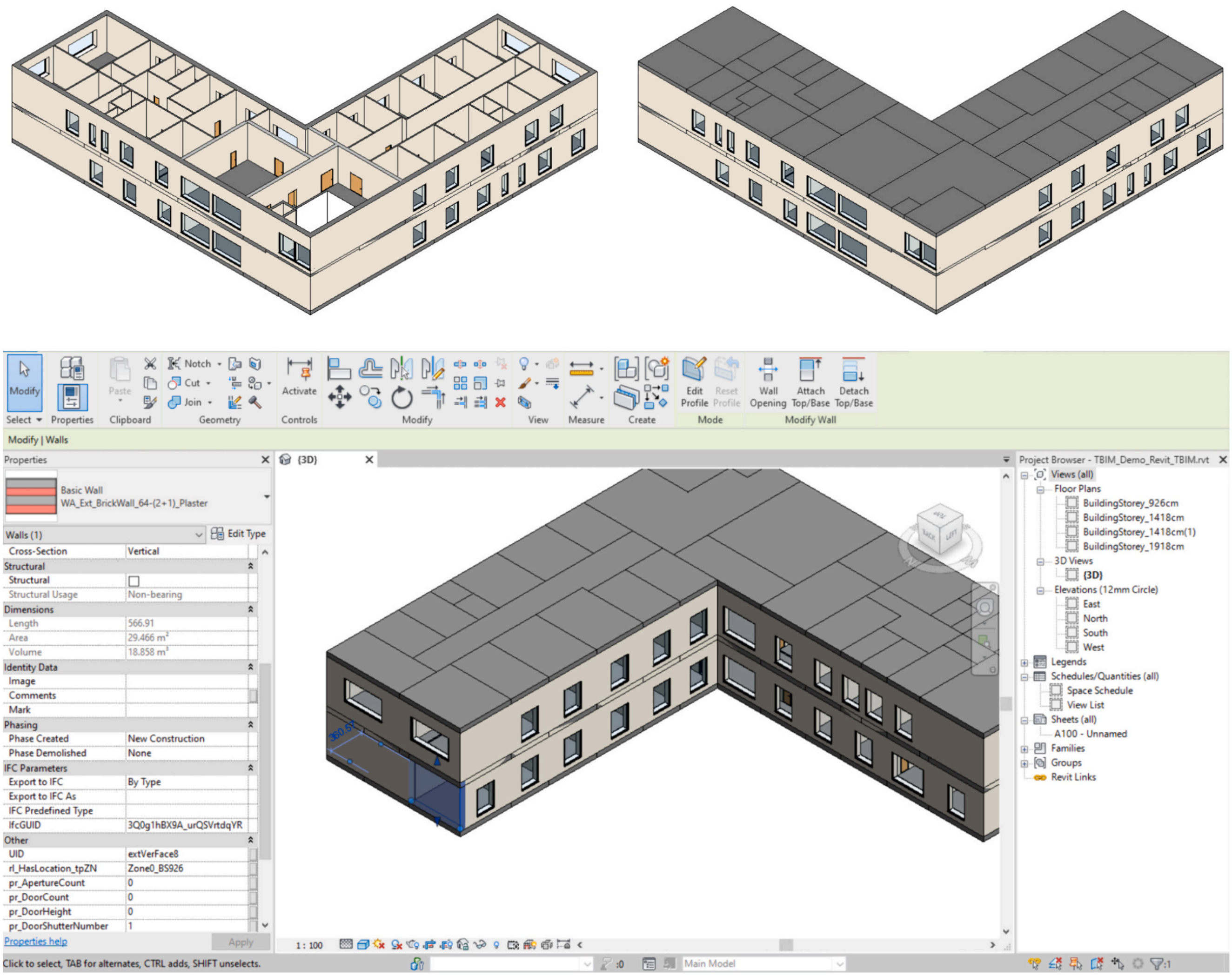


Fig. 14. BIM model in Autodesk Revit.

4.6. BEM modeling

Finally, the Topologic TBIM was processed to generate the BEM and run the energy analysis. This operation was carried out using an approach similar to the Topologic-Revit conversion but aligning Topologic's element classes and properties with those of Ladybug Tools, specifically HB, through the Ladybug Tools APIs in Python.

First, Topologic cells were aggregated into Topologic clusters according to their thermal and occupancy data. These clusters represented aggregations of spaces intended as energy zones of the buildings to be analyzed through energy analysis. The zoning operation leveraged the 'Cluster.K_Means' method from Topologicpy, which allowed the cells to

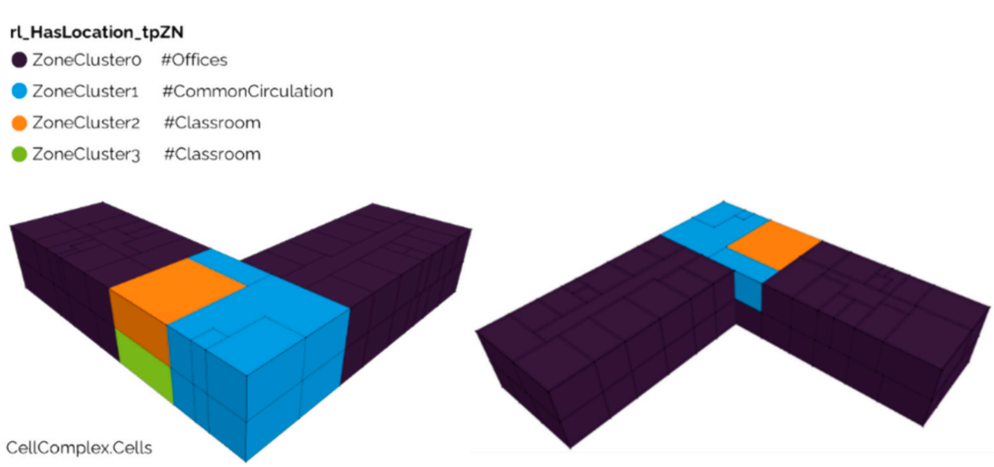


Fig. 15. Energy zones identified by the K-Means algorithm.

be clustered based on some of their properties. In this case, adjacent cells were grouped into zones by inputting 'AreaPerOccupancyPeak', 'IsHeated', and 'IsOccupied' properties in the K-Means clustering algorithm, resulting in the zones depicted in Fig. 15.

Subsequently, the properties of the cells were transferred to the clusters through aggregation operations such as sum, average, weighted average, minimum, and maximum, depending on the properties transferred. For example, the maximum number of people in the area was calculated by summing the maximum number of people in each space within the zone. Alternatively, the equipment and lighting power density was determined by taking the weighted average of the values across the spaces. This process was repeated for other parameters previously described.

Then, the Topologic faces were converted into HB wall, roof, and floor surfaces by integrating Topologicpy with the HB APIs, and relevant material values were transferred to create HB constructions [67]. Similarly, the Topologic apertures were converted into HB window surfaces, transferring data such as U-value, solar heat gain coefficient, and visual transmittance to get the corresponding HB glazed constructions. Once the basic geometries of the HB model were established, the HB zones were created, and the energy data from the respective Topologic clusters were transferred to them. In particular, the energy loads were set thanks to the 'honeybee_energy.load' package, while constructions were converted thanks to the 'honeybee_energy.construction' module.

The outcome was a BEM model in the Honeybee JSON format. This model was then converted into an IDF model using the 'honeybee.model' module. Additionally, within PyRevit, the Eppy library was used to modify the file by assigning the operational schedules of the spaces to the respective zones, which were manually created in a format compliant with EnergyPlus. The resulting model was inputted into EnergyPlus, along with the EnergyPlus weather climate file of Bologna Borgo Panigale, to conduct energy simulations and calculate zone by zone parameters such as cooling, heating, and lighting energy needs. After the simulation, the resulting energy data was saved in the EP's output CSV format, linked to zones of the TBIM by enriching the clusters' dictionaries, and displayed on the TBIM geometries, as shown in Fig. 16. To validate this passage, a GH script for BIM to BEM translation, tested in previous research [62,68], was applied to the case study's Revit model to reproduce the BEM.

5. Discussion

5.1. Process evaluation

The main methodological steps followed in the research are evaluated here, considering factors such as the degree of automation, the required inputs, the computational complexity, and the extensibility to new applications.

Regarding the topology modeling phase, Topologicpy has proven to be particularly effective in enabling the modeling of the topological aspects of existing buildings and framing them within a space-centric perspective. On the one hand, the theoretical principles behind this software helped set up the theoretical framework and the conceptual workflow of the study. On the other hand, the Topologicpy toolkit enabled the practical application of topology modeling principles, functioning as the core of the toolchain described in the article. The great advantage of the topology-based approach to BIM lied in using models with a low level of geometric detail, modeled for spaces rather than construction components and, for this reason, compatible with the kernels of energy simulation software. The computational complexity of these models was exceptionally low, allowing the whole process to run efficiently on standard-performance computers. The only necessary input for the topology modeling process included a 3D model representing the spaces of the case study building as gross closed volumes. This model was realized through simple CAD retracing operations, which did not require much time or high specialization to be executed. Even though 3D modeling was tackled manually in the study – an aspect that contrasts the semi-automation principle proposed in the theoretical framework – it is worth noting that, with the increasing focus on AI research in the AECO sector, updates to the method for automating the 3D modeling process can be quickly expected, involving algorithms for the segmentation and recognition of objects in the floor plans and the automated retracing of boundary curves for spaces.

The information enrichment step significantly contributed to the semi-automation of the TBIM process. In particular, rule-based information modeling techniques were proven helpful for automatically enriching the models of BPS-related data through semantic and topological queries. The use of the ILDs to attribute new data to spaces has proven to be perfectly aligned with the proposed space-oriented vision of BIM, since all information about the building's occupancy and energy requirements was attributed based on the functions of the spaces, placing, therefore, central importance on spatial elements for performing information enrichment. This aspect also aligns with the typical approach to building management, often focused on the concept of space for programming occupancy and analyzing occupants' needs. Furthermore, the use of the IRSs was particularly effective in adding construction data to the face and aperture elements of the models.

In summary, the application presented showed how conditional modeling can be used not only for model checking, which is the current focus of most BIM research, but also for model enrichment tasks. This approach could also be replicated for designing new buildings, where the performance characteristics of the interface elements are often tied to the functional requirements of the spaces they enclose. Moreover, it could be repeated similarly for performance analyses different from energy simulations, such as fire safety, daylighting, thermal comfort, or acoustics.

Concerning the BIM modeling step, the main result of this research is the definition of the so-called TBIM. This conceptual informative model is geometrically simple but semantically structured, from which more detailed models can be derived for specific informational uses, such as energy simulations.

The effectiveness of the TBIM models for generating the derived models lied in three main aspects. Firstly, the format of the TBIM (the Topologic JSON format) was straightforward and easily manageable. This aspect facilitated the development of the PyRevit algorithms with

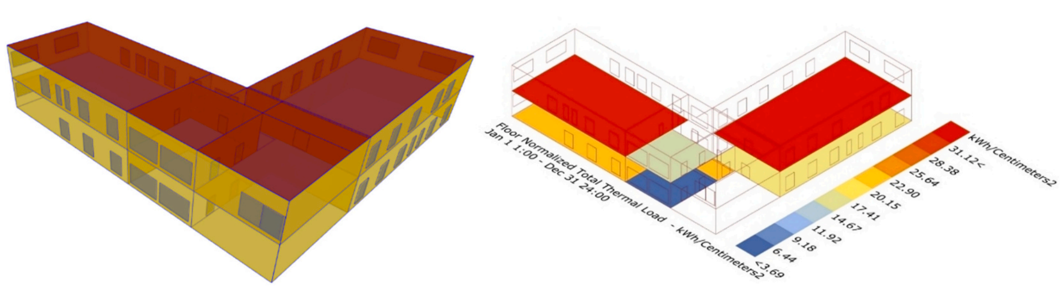


Fig. 16. Energy model. On the left, IDF model. On the right, results from energy analysis (total thermal load).

relative ease, particularly compared to the algorithms that would have been required to develop the toolkit with more complex data schemes, such as native IFC. Secondly, the topological organization of the model allowed for the standardization of the information added to it. Moreover, it secured algorithmic control of the modeling procedure, limiting the errors often caused by human data entry in traditional BIM procedures and ensuring the semantic integrity of information. Indeed, rather than checking the data after it has been added, the presented process ensured that it was correctly added from the beginning. Third, being easily convertible into a graph network, the TBIM model gained all the benefits usually associated with graph knowledge structure. On the one hand, the use of graphs allowed for organizing information structure in more complex forms than other forms common in traditional BIM, such as relational databases. On the other hand, the graph system allowed for the scalability of the process towards other applications. It is straightforward to hypothesize how the structure of the TBIM could be aligned with ontologies such as BOT and Brick to share building data on the Internet (through JSON-LD or RDF formats), a requisite increasingly demanded in the building management sector for developing web-based smart building and DT applications.

Finally, regarding the BPS modeling stage, the proposed method allowed for significant interoperability between BIM and BEM, producing geometrically simple models with high informational content, easily processable in BPS software. Rather than being rebuilt from scratch, the BEM model was derived from the TBIM model by aligning it with the notations typical of established BEM environments, namely HB and EP. This alignment occurred from an ontological and content perspective, enabling the organic transfer of the data necessary for energy analysis from the TBIM to the BEM. The efforts required by the modeler in the initial steps to adhere to the method's rigor, particularly from a semantic standpoint, were rewarded in this phase when the BEM generation process was automated. This adds value to the whole process, drastically reducing BPS modeling time and enabling reliance on the same model for multiple purposes. In this case, it is easy to foresee how the approach presented could be extended to design new buildings, where energy models should serve the early-stage phase to allow designers to grasp the energy impacts of design choices.

In general, applying the approach proved to be aligned with the research objectives. Although there are areas for technical improvement, as outlined in the following paragraph, the tools used demonstrated their utility in enabling BIM modeling focused on topological concepts and the transition from product-based to space-based information modeling in existing buildings. Additionally, the approach proved particularly effective in producing BIM models compatible with BPS. Therefore, both the objectives of the research – space-oriented and BPS-compatible BIM – were successfully achieved.

5.2. Limitations and future work

The case study presented was specifically chosen as a demonstrative compromise of the approach presented in the paper, being neither too small nor too large, simple from a geometric point of view but highly complex from a functional standpoint. Moreover, the construction and architectural characteristics of the building chosen for the demonstration – an Italian rationalist building with a massive character – were not particularly difficult to model through the presented method. The modeling process of the case study did not encounter any particular obstacles; however, it is necessary to consider that the presented toolkit exhibits several technical limitations that require further development to make it suitable for more complex contexts.

First, the toolkit presented only allowed handling prism geometries. Unlike Topologicpy and Energy Plus, which can handle a more extensive variety of planar shapes, the toolkit struggled with more complex architectural forms involving inclined or curved surfaces. This limitation was solely due to the step when the translation of the Topologic TBIM model into the Autodesk Revit BIM model was performed (Section 4.5). The resolution of this gap is purely technical and not theoretical; it would be sufficient to adapt the methods provided by the Autodesk Revit APIs for other types of geometry, accommodating the possibility of modeling pitched roofs and slanted walls, both functionalities enabled by the latest updates to Revit's APIs (2024).

Second, the computational performance of the toolkit heavily depended on the size of the building. The modeling process was quite streamlined for modeling the case study, which comprises about 65 cells and 330 faces. Considering that this led to the joint generation of a BIM model and a BEM model, it is reasonable to estimate that applying the method at least halved the modeling times compared to the traditional BIM and BEM approaches. However, in scenarios where buildings are larger (which means they contain more spaces), the toolkit's performance may suffer. A main computational bottleneck may occur in one of the first modeling steps, when the Collector Model cell complex is generated from the BRep files (Section 4.3). Fig. 17 illustrates the increase in time required for this operation as the number of cells and faces belonging to the cell complex grows. On the one hand, it is worth noting that larger buildings would necessitate longer cell complex generation times. On the other, it should be acknowledged that an ordinary laptop (CPU 2,60 GHz, RAM 16GB) was used for this research; therefore, an increase in computational resources is expected to improve performance. Furthermore, it should be noted that Topologicpy 0.4.30 version was used for this work. The latest updates to this software library have made topology modeling operations faster, so the computational performance issues could be resolved.

Although based on several open-source software libraries (i.e., Topologic, LadyBug Tools, EP), the toolkit operates within Autodesk Revit, which, despite being used by many AECO professionals, is a native closed BIM environment. Moreover, it relies on Rhino and Grasshopper for some 3D and BEM modeling procedures. Addressing these issues through enhancements and refinements will be essential to maximize the utility of TBIM and support its potential impact in the building management field. An open, flexible authoring environment would allow for better compatibility and expand the toolkit's accessibility in various workflows, as done by Postle in BlenderBIM [43]. In particular, using Autodesk Revit and PyRevit could be avoided by developing a Python-based GUI. Indeed, in this study, Revit was used only for user-model interaction.

Fourth, in the BEM, HVAC systems were not modeled. Therefore, the toolkit applies solely to calculating the energy needs of spaces rather than actual consumption. For this reason, the workflow would be expanded to accommodate this aspect. A possible way to execute this extension would include modeling HVAC equipment elements in the TBIM model as Topologic vertices, linking them topologically to spatial elements through containment queries, and converting them into IDF elements via the Ladybug Tools APIs. Brick can be particularly helpful for mapping items in the topology model; thus, adding it to the federation of ontologies (described in Section 3.2.1) would be beneficial to enhance semantically this procedure.

Fifth, adopting the proposed method necessitated advanced digital and data flow management skills to ensure standardized procedures and a comprehensive understanding of BIM and BEM techniques. In particular, VP expertise is required during the transition from BIM to BEM when BIM models are already available for building management operators, and there is no need to produce new BIMs passing from TBIMs. Moreover, individuals proficient in BIM may lack the specialized knowledge required for energy modeling in EP, which is considerably more complicated than the energy performance methods commonly used by professionals (for instance, compared to the methods within the Italian building energy performance certification system [69]). However, as both knowledge bases and technical fields are continually evolving, it can be anticipated that skills will become increasingly prevalent in the industry [30].

Lastly, with regard to the studies discussed in the first part of the text, it is crucial to consider that combining the suggested method with

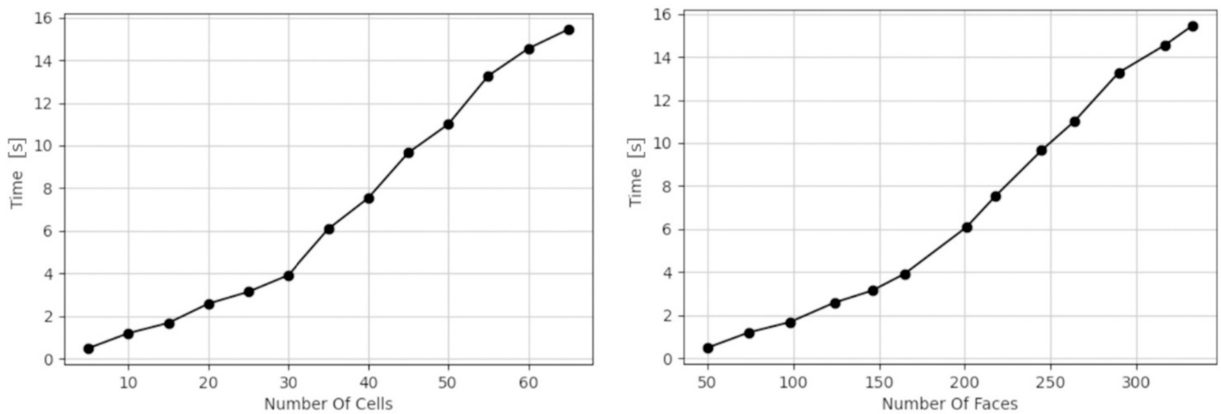


Fig. 17. Time required for the generation of cell complexes based on the number of cells and faces.

automated floor plan object recognition and automated space boundary curve retracing – possibly via AI – would accelerate model generation even further while guaranteeing semantic and geometric compatibility with BPS.

6. Conclusion

With the increasing digitization of the construction sector, new digital tools are required to support decision-making in building management, aiming to maintain a sustainable, high-performance, and energy-efficient built environment. Despite significant progress, there are still barriers limiting the adoption of digital technologies, such as BIM and BPS, in the field of building management. This is primarily due to the high skills required to adopt these technologies, the technical difficulties in evaluating building performance, the substantial economic and technical resources needed to digitize large real estate assets, and the lack of standardization related to performance from a management perspective.

Improving educational programs of higher education institutions – which are in charge of transferring new digital and specialized skills to the industry to assist its digital and ecological transformation – seems to be a viable way to address the first two problems. The BIM research community, on the other hand, is credited with helping to overcome the remaining challenges in two ways: first, by creating new approaches targeted at automating workflows and data acquisition with techniques like automatic scanning, semantic segmentation and modeling; and second, by integrating these methods into more comprehensive approaches that address the challenge of digitalization by moving towards information systems designed to meet specific objectives and systematically integrated with performance assessment strategies.

This article, aiming to overcome a subset of these gaps, has introduced an approach termed Topological Building Information Modeling with two main objectives. Firstly, it sought to semi-automate the BIM processes for existing buildings, reducing the time and resources required to create such models compared to traditional approaches. The second objective was to standardize BIM modeling procedures to produce consistent models highly interoperable with BPS environment – specifically focusing on energy-related aspects – usable for conducting energy analysis in support of performance-based building management. These models can be used to derive more detailed BIM models or models for specific information uses, such as BEM. The TBIM models are generated semi-automatically and semantically standardized. Although they are built with low geometric detail, they have high informational content. Both these latter aspects make them interoperable with external BPS applications.

The practical outcome of the research is a software toolkit formalized based on a theoretical framework centered around the concept of space-oriented and topology-based BIM. This toolkit was developed in Python and PyRevit and uses Topologicpy as the core for modeling building topology and Autodesk Revit as the primary BIM environment. Future developments would make it possible to extend the toolkit to open BIM platforms with simple code adjustments in the last part of the method (when the Topologic TBIM model is transformed into an Autodesk Revit BIM model). Additionally, further work would involve adopting graph machine learning tools to replace or integrate rule-based information enrichment algorithms, further automating operations related to the semantic enrichment of the models.

The added contribution of the paper, therefore, concerns not only the formalization of a toolchain for semi-automated BIM modeling but, more broadly, of an approach to the informational modeling of buildings, here defined as TBIM, which transitions product-oriented to space-centered BIM.

CRediT authorship contribution statement

Angelo Massafra: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Wassim Jabi:** Writing – review & editing, Validation, Supervision, Software, Project administration, Methodology, Conceptualization. **Riccardo Gulli:** Writing – review & editing, Validation, Supervision, 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.

Data availability

A public GitHub repositoy has been created for sharing data and models presented in the paper

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] A.I. AbdelAzim, A.M. Ibrahim, E.M. Aboul-Zahab, Development of an energy efficiency rating system for existing buildings using analytic hierarchy process – the case of Egypt, Renew. Sust. Energ. Rev. 71 (2017) 414–415, https://doi.org/10.1016/j.rser.2016.12.071.

[2] Ministero dell''Economia e delle Finanze (MEF), Patrimonio della PA. Rapporto tematico. Modelo di stima del valore del patrimonio immobiliare pubblico, Available, https://www.dt.mef.gov.it/export/sites/sitodt/modules/documenti_it

/programmi_cartolarizzazione/patrimonio_pa/Modelo_Stima_Valore_Immobili_Pubblici.pdf, 2015 [Accessed: Mar. 15, 2022].

[3] Italian Legislative Decree of January 22, 2004, No. 42. Cultural Heritage and Landscape Code, pursuant to Article 10 of Law July 6, 2002, No. 137/2004, Available: https://www.gazzettaufficiale.it/anteprima/codici/beniCulturali, 2024 [Accessed Jul. 5, 2024].

[4] R. Bortolini, N. Forcada, Analysis of building maintenance requests using a text mining approach: building services evaluation, Build. Res. Inf. 48 (2) (2020) 207–217, https://doi.org/10.1080/09613218.2019.1609291.

[5] I. Petri, Y. Rezgui, A. Ghoroghi, A. Alzahrani, Digital twins for performance management in the built environment, J. Ind. Inf. Integr. 33 (2023) 100445, https://doi.org/10.1016/j.jii.2023.100445.

[6] M. Deng, C.C. Menassa, V.R. Kamat, From BIM to digital twins: a systematic review of the evolution of intelligent building representations in the AEC-FM industry, J. Inform. Technol. Construct. 26 (2021) 58–83, https://doi.org/10.36680/j.itcon.2021.005.

[7] X. Gao, P. Pishdad-Bozorgi, BIM-enabled facilities operation and maintenance: a review, Adv. Eng. Inform. 39 (2019) 227–247, https://doi.org/10.1016/j.aei.2019.01.005.

[8] M. Murphy, E. McGovern, S. Pavia, Historic building information modeling (HBIM), Struct. Surv. 27 (4) (2009) 311–327, https://doi.org/10.1108/02630800910985108.

[9] P. de Wilde, Building performance simulation in the brave new world of artificial intelligence and digital twins: a systematic review, Energ. Build. 292 (2023) 113171, https://doi.org/10.1016/j.enbuild.2023.113171.

[10] M.S. Aliero, M. Asif, I. Ghani, M.F. Pasha, S.R. Jeong, Systematic review analysis on smart building: challenges and opportunities, Sustainability 14 (5) (2022) 3009, https://doi.org/10.3390/su14053009.

[11] M. Almatared, H. Liu, S. Tang, M. Sulaiman, Z. Lei, H.X. Li, Digital twin in the architecture, engineering, and construction industry: a bibliometric review, in: Proceedings of Construction Research Congress 2022, Arlington, Virginia: American Society of Civil Engineers, 2022, pp. 670–678, https://doi.org/10.1061/9780784483961.070.

[12] F. Zhang, A.P.C. Chan, A. Darko, Z. Chen, D. Li, Integrated applications of building information modeling and artificial intelligence techniques in the AEC/FM industry, Autom. Constr. 139 (Jul. 2022) 104289, https://doi.org/10.1016/j.autcon.2022.104289.

[13] C. Boje, A. Guerriero, S. Kubicki, Y. Rezgui, Towards a semantic construction digital twin: directions for future research, Autom. Constr. 114 (2020) 103179, https://doi.org/10.1016/j.autcon.2020.103179.

[14] X. Xie, N. Moretti, J. Merino, A.K. Parlikad, Ontology-based spatial and system hierarchies federation for fine-grained building energy analysis, in: Proceedings of 38th International Conference of CIB W78, Luxembourg, Oct. 2021. Available: https://www.researchgate.net/publication/355104659_Ontology-Based_Spatial_and_System_Hierarchies_Federation_for_Fine-Grained_Building_Energy_Analysis [Accessed: Feb. 27, 2024].

[15] P. de Wilde, Building Performance Analysis, Wiley Blackwell, Hoboken, NJ, 2018. ISBN 9781119341925.

[16] T. Abuimara, B.W. Hobson, B. Gunay, W. O'Brien, M. Kane, Current state and future challenges in building management: practitioner interviews and a literature review, J. Build. Eng. 41 (2021) 102803, https://doi.org/10.1016/j.jobe.2021.102803.

[17] B. Bortoluzzi, I. Efremov, C. Medina, D. Sobieraj, J.J. McArthur, Automating the creation of building information models for existing buildings, Autom. Constr. 105 (2019) 102838, https://doi.org/10.1016/j.autcon.2019.102838.

[18] R. Romero-Jarén, J.J. Arranz, Automatic segmentation and classification of BIM elements from point clouds, Autom. Constr. 124 (2021) 103576, https://doi.org/10.1016/j.autcon.2021.103576.

[19] S. Chen, R. Jin, M. Alam, Investigation of interoperability between building information modeling (BIM) and building energy simulation (BES), Int. Rev. Appl. Sci. Eng. 9 (2) (2018) 137–144, https://doi.org/10.1556/1848.2018.9.2.9.

[20] C.M. Eastman, Building Product Models: Computer Environments Supporting Design and Construction, CRC Press, Boca Raton, Fla, 1999. ISBN 9780849302596.

[21] Y. Yang, Y. Pan, F. Zeng, Z. Lin, C. Li, A gbXML reconstruction workflow and tool development to improve the geometric interoperability between BIM and BEM, Buildings 12 (2) (2022) 221, https://doi.org/10.3390/buildings12020221.

[22] G.B. Porsani, K. Del Valle, A. de Lersundi, Sanchez-Ostiz Gutièrrez, C.F. Bandera, Interoperability between building information modeling (BIM) and building energy model (BEM), Appl. Sci. 11 (5) (2021) 2167, https://doi.org/10.3390/app11052167.

[23] Topologicpy, Available, https://pypi.org/project/topologicpy/, 2024 [Accessed: May 25, 2023].

[24] LadyBug Tools, Available, https://www.ladybug.tools/, 2024 [Accessed: Jan. 18, 2023].

[25] EnergyPlus, Available, https://energyplus.net/, 2024 [Accessed: May 08, 2023].

[26] A. Massafra, TBIM, Available: https://github.com/angelomassafra/TBIM, 2023 [Accessed: Feb 27, 2024].

[27] V. Pătrăucean, I. Armeni, M. Nahangi, J. Yeung, I. Brilakis, C. Haas, State of research in automatic as-built modeling, Adv. Eng. Inform. 29 (2) (Apr. 2015) 162–171, https://doi.org/10.1016/j.aei.2015.01.001.

[28] M. Weinmann, B. Jutzi, S. Hinz, C. Mallet, Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers, ISPRS J. Photogramm. Remote Sens. 105 (2015) 286–304, https://doi.org/10.1016/j.isprsjprs.2015.01.016.

[29] E. Valero, A. Forster, F. Bosché, E. Hyslop, L. Wilson, A. Turmel, Automated defect detection and classification in ashlar masonry walls using machine learning, Autom. Constr. 106 (2019) 102846, https://doi.org/10.1016/j.autcon.2019.102846.

[30] C.-S. Hsieh, X.-J. Ruan, Automated semantic segmentation of indoor point clouds from close-range images with three-dimensional deep learning, Buildings 13 (2) (2023) 468, https://doi.org/10.3390/buildings13020468.

[31] J. Collao, F. Lozano-Galant, J.A. Lozano-Galant, J. Turmo, BIM visual programming Tools applications in infrastructure projects: a state-of-the-art review, Appl. Sci. 11 (18) (2021) 8343, https://doi.org/10.3390/app11188343.

[32] S. Yang, M. Hou, S. Li, Three-dimensional point cloud semantic segmentation for cultural heritage: a comprehensive review, Remote Sens. 15 (3) (2023) 548, https://doi.org/10.3390/rs15030548.

[33] F. Bosché, M. Ahmed, Y. Turkan, C.T. Haas, R. Haas, The value of integrating scan-to-BIM and scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: the case of cylindrical MEP components, Autom. Constr. 49 (2015) 201–213, https://doi.org/10.1016/j.autcon.2014.05.014.

[34] J.S. Lee, J. Park, Y.-M. Ryu, Semantic segmentation of bridge components based on hierarchical point cloud model, Autom. Constr. 130 (2021) 103847, https://doi.org/10.1016/j.autcon.2021.103847.

[35] C. Yin, B. Wang, V.J.L. Gan, M. Wang, J.C.P. Cheng, Automated semantic segmentation of industrial point clouds using ResPointNet++, Autom. Constr. 130 (2021) 103874 https://doi.org/10.1016/j.autcon.2021.103874.

[36] C. Zhang, Y. Zou, J. Dimyadi, A systematic review of automated BIM modeling for existing buildings from 2D documentation, in: Proceedings of 38th International Symposium on Automation and Robotics in Construction, UAE, Dubai, 2021, https://doi.org/10.22260/ISARC2021/0032.

[37] Q. Lu, L. Chen, S. Li, M. Pitt, Semi-automatic geometric digital twinning for existing buildings based on images and CAD drawings, Autom. Constr. 115 (2020) 103183, https://doi.org/10.1016/j.autcon.2020.103183.

[38] B. Yang, B. Liu, D. Zhu, B. Zhang, Z. Wang, K. Lei, Semiautomatic structural BIM-model generation methodology using CAD construction drawings, J. Comput. Civ. Eng. 34 (3) (2020) 04020006, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000885.

[39] M. Pritoni, et al., Metadata schemas and ontologies for building energy applications: a critical review and use case analysis, Energies 14 (7) (2021) 2024, https://doi.org/10.3390/en14072024.

[40] buildingSMART International, International Foundation Classes (IFC), Available, https://www.buildingsmartitalia.org/standard/standard-bs/industry-foundation-classes-ifc/ [Accessed: Aug. 29, 2021].

[41] M.H. Rasmussen, P. Pauwels, C.A. Hviid, J. Karlshøj, Proposing a central AEC ontology that allows for domain specific extensions, in: Proceedings of the Joint Conference on Computing in Construction, Heraklion, Crete, Greece: Heriot-Watt University, Jul. 2017, pp. 237–244, https://doi.org/10.24928/JC3-2017/0153.

[42] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Bergés, D. Culler, R.K. Gupta, M.B. Kjærgaard, M. Srivastava, K. Whitehouse, Brick : metadata schema for portable smart building applications, Appl. Energy 226 (2018) 1273–1292, https://doi.org/10.1016/j.apenergy.2018.02.091.

[43] B. Postle, An adaptive approach to domestic design, Figshare Contrib. (2018), https://doi.org/10.6084/m9.figshare.6267401.v1.

[44] M. Villegas Ballesta, Informational Architecture: a Postgraphical Knowledge Model for Architectural Design, PhD Thesis, Universidad de Sevilla, 2022, https://idus.us.es/handle/11441/138108 [Accessed: Mar. 21, 2023].

[45] W. Jabi, A. Chatzivasileiadi, Topologic: Exploring spatial reasoning through geometry, topology, and semantics, in: S. Eloy, D. Leite Viana, F. Morais, J. Vieira Vaz (Eds.), Formal Methods in Architecture. Advances in Science, Technology & Innovation, Springer, Cham, 2021, https://doi.org/10.1007/978-3-030-57509-0_25.

[46] P. Janssen, K.W. Chen, A. Mohanty, Automated generation of BIM models, in: Proceedings of eCAADe 2016: Complexity & Simplicity, Oulu, Finland, 2016, pp. 583–590, https://doi.org/10.52842/conf.ecaade.2016.2.583.

[47] BuildingSMART Italy, Guideline on the use of BIM for energy performance, in: Which Information is Exchanged, When, Why, and among Who?, 2021. Available, https://www.buildingsmartitalia.org/utenti/pubblicazioni/energy-performance/ [Accessed: Oct. 28, 2021].

[48] M. Bonomolo, S. Di Lisi, G. Leone, Building information modeling and energy simulation for architecture design, Appl. Sci. 11 (5) (2021) 2252, https://doi.org/10.3390/app11052252.

[49] V. Panagiotidou, A. Korner, From Intricate to Coarse and Back. A voxel-based workflow to approximate high-res geometries for digital environmental simulations, in: Proceedings of eCAADe2023 Co-creating the Future: Inclusion in and through Design, Ghent, Belgium, 2022, https://doi.org/10.52842/conf.ecaade.2022.1.491.

[50] G. Costa, A. Sicilia, Web technologies for sensor and energy data models, in: Building and Semantics. Data Models and Web Technologies for the Built Environment, CRC Press/Balkema, Leiden, The Netherlands, 2023. ISBN 9781032023120.

[51] E. Kamel, A.M. Memari, Review of BIM's application in energy simulation: Tools, issues, and solutions, Autom. Constr. 97 (2019) 164–180, https://doi.org/10.1016/j.autcon.2018.11.008.

[52] C. Mazzoli, M. Iannantuono, V. Giannakopoulos, A. Fotopoulou, A. Ferrante, S. Garagnani, Building information modeling as an effective process for the sustainable re-shaping of the built environment, Sustainability 13 (9) (2021) 4658, https://doi.org/10.3390/su13094658.

[53] L.C. Tagliabue, S. Maltese, F.R. Cecconi, A.L.C. Ciribini, E. De Angelis, BIM-based interoperable workflow for energy improvement of school buildings over the life cycle, in: Proceedings of 34th International Symposium on Automation and

Robotics in Construction, Taipei, Taiwan, 2018, https://doi.org/10.22260/ISARC2018/0058.
[54] E. Kamel, A. Kazemian, BIM-integrated thermal analysis and building energy modeling in 3D-printed residential buildings, Energ. Build. 279 (2023) 112670, https://doi.org/10.1016/j.enbuild.2022.112670.
[55] S. Jiang, X. Feng, B. Zhang, J. Shi, Semantic enrichment for BIM: enabling technologies and applications, Adv. Eng. Inform. 56 (2023) 101961, https://doi.org/10.1016/j.aei.2023.101961.
[56] H. Ying, S. Lee, A rule-based system to automatically validate IFC second-level space boundaries for building energy analysis, Autom. Constr. 127 (2021) 103724, https://doi.org/10.1016/j.autcon.2021.103724.
[57] W. Solihin, C. Eastman, Classification of rules for automated BIM rule checking development, Autom. Constr. 53 (2015) 69–82, https://doi.org/10.1016/j.autcon.2015.03.003.
[58] J. Merino, X. Xie, N. Moretti, J.Y. Chang, A.K. Parlikad, Data integration for digital twins in the built environment, in: Proceedings of 2022 European Conference on Computing in Construction, Jul. 2022, https://doi.org/10.35490/EC3.2022.172.
[59] L. Chamari, E. Petrova, P. Pauwels, A web-based approach to BMS, BIM and IoT integration, in: Proceedings of CLIMA 2022 14th REHVA HVAC World Congress, May 2022, https://doi.org/10.34641/CLIMA.2022.228.
[60] Z. Wang, R. Sacks, T. Yeung, Exploring graph neural networks for semantic enrichment: room type classification, Autom. Constr. 134 (2022) 104039, https://doi.org/10.1016/j.autcon.2021.104039.
[61] PyRevit, Available, https://github.com/eirannejad/PyRevit, 2024 [Accessed: Apr. 05, 2023].
[62] A. Massafra, R. Gulli, Enabling bidirectional interoperability between BIM and BPS through lightweight topological models, in: Proceedings of eCAADe 2023: Digital Design Reconsidered, Graz, Austria, 2023, pp. 187–196, https://doi.org/10.52842/conf.ecaade.2023.2.187.
[63] Rhino.Inside.Revit, m, https://www.rhino3d.com/inside/revit/beta/, 2024 [Accessed: Apr. 05, 2023].
[64] G. Predari, D. Prati, A. Massafra, Modern construction in Bologna. The Faculty of Engineering by Giuseppe Vaccaro, 1932–1935, in: Digital Modernism Heritage Lexicon, Springer Nature Switzerland AG, Cham, Svizzera, 2021, pp. 233–258, https://doi.org/10.1007/978-3-030-76239-1_11.
[65] Robert McNell & Associates, Rhino - Rhinoceros 3D, Available, https://www.rhino3d.com/it/, 2024 [Accessed: Apr. 30, 2024].
[66] Revit API Docs, Online documentation for the Revit APIs, Available, https://www.revitapidocs.com/, 2024 [Accessed: Apr. 30, 2024].
[67] Ladybug Tools, API Docs - Honeybee Energy Documentation, Available: https://www.ladybug.tools/honeybee-energy/docs/api.html, 2024.
[68] A. Massafra, C. Costantino, G. Predari, R. Gulli, Building information modeling and building performance simulation-based decision support Systems for Improved Built Heritage Operation, Sustainability 15 (14) (2023) 11240, https://doi.org/10.3390/su151411240.
[69] Ministro dello Sviluppo Economico, Ministro dell''Ambiente e della Tutale del Territorio e del Mare, Ministro delle Infrastrutture e dei Trasporti, and Ministro per la Semplificazione e la Pubblica Amministrazione, Decreto interministeriale 26 giugno 2015 - Adeguamento linee guida nazionali per la certificazione energetica degli edifici, Available: https://www.gazzettaufficiale.it/eli/id/2015/07/15/15A05200/sg, 2015 [Accessed Jul. 5, 2024].