



Graph-Based methodology for Multi-Scale generation of energy analysis models from IFC



Asier Mediavilla ^{a,*}, Peru Elguezabal ^{a,b}, Natalia Lasarte ^a

^a TECNALIA, Basque Research and Technology Alliance (BRTA), Derio, Spain

^b Department of Electrical Engineering, University of the Basque Country UPV/EHU, Bilbao, Spain

ARTICLE INFO

Article history:

Received 11 August 2022

Revised 22 December 2022

Accepted 8 January 2023

Available online 11 January 2023

Keywords:

Open BIM

BEM

Multiscale models

Building graphs

Process automation

IFC

gbXML

Energy Plus

ABSTRACT

Process digitalisation and automation is unstoppable in all industries, including construction. However, its widespread adoption, even for non-experts, demands easy-to-use tools that reduce technical requirements. BIM to BEM (Building Energy Models) workflows are a clear example, where ad-hoc prepared models are needed. This paper describes a methodology, based on graph techniques, to automate it by highly reducing the input BIM requirements found in similar approaches, being applicable to almost any IFC. This is especially relevant in retrofitting, where reality capture tools (e.g., 3D laser scanning, object recognition in drawings) are prone to create geometry clashes and other inconsistencies, posing higher challenges for automation. Another innovation presented is its multi-scale nature, efficiently addressing the surroundings impact in the energy model. The application to selected test cases has been successful and further tests are ongoing, considering a higher variety of BIM models in relation to tools and techniques used and model sizes.

© 2023 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

1.1. Context and interest of the research

In the pursue of the challenging decarbonisation goals and the need to reach energy efficient buildings, one key driver is to deeply understand the thermal behaviour of a building and the impact that the surroundings have on it. The EU has established a roadmap to achieve the energy and environmental goals [1] for the whole continent, and, for the case of the building sector, the target is to transform it into a highly energy efficient and decarbonised stock by 2050. The Energy Performance of Buildings Directive (EPBD) is the legal framework that has been adopted to guide the process, a document that was first published in 2002 [2] and that has been continuously reviewed and updated in recent years, where the relevance and impact of renovation activities has been continuously growing. The last review is planned to be published soon and in the draft version of this new regulation [3], the necessity to focus the efforts on those buildings with higher improvement possibilities is highlighted.

But, although the impact of different technologies attending to their conventional performance can be estimated [4], the design and definition of renovation projects to achieve high energy efficiency levels require individual and detailed analysis for every specific building, considering passive measures [5] (insulation, windows, air permeability) combined with active technologies [6] (renewables [7], HVAC system). Simulation has been widely adopted to predict alternative renovation projects combining different technology packages, by means of estimating the final energy that could be needed to operate the building once retrofitted [8].

On the other hand, the digitalisation and automation of the construction sector is becoming a driver to transform and improve the productivity of one of the least profitable industries [9,10]. This statement is, in fact, gathered in certain European directives, aimed at boosting the use of digital tools in the building sector [11]. The use of BIM, as an emerging technology for the digitalisation of the sector, is intended to solve the inefficiencies of the construction and renovation process while increasing collaboration between stakeholders [12–14].

Given the fact that, having a BEM model enables the energy simulation which leads to the design of energy efficient buildings, leveraging the potential of BIM also for this purpose becomes essential. The BIM model of the new or existing building can be used as inputs for BEM, rather than energy modelling from scratch,

* Corresponding author.

E-mail address: asier.mediavilla@tecnalia.com (A. Mediavilla).

optimising BIM to BEM processes, minimizing errors and promoting collaboration between stakeholders.

The increasing adoption of digitalisation in construction and stricter energy efficiency regulations imply that the following desirable objectives could potentially become mandatory in the future, especially for public funded projects:

- Make energy performance simulation a built-in aspect of any building life-cycle process (early design, detailed design, or operation) and both for new buildings and renovation of existing ones.
- Make the process as automatic as possible, to avoid human modelling errors and reduce input requirements.
- Consider together the building geometry with composing materials, as well as the building context, especially for cross-effects (shadowing, wind...).
- Rely on open standards, to avoid software vendor-dependency, thus making it affordable to any stakeholder profile, big companies to SMEs.

In some use-cases, such as detailed building design, a very high-definition level of the building and energy parameters is needed, with expert analysts involved, but for some other scenarios (e.g., decision-making processes at early design stage), the energy model can have a lower detail, because only a comparative analysis of alternatives is expected. For those scenarios an additional objective should be pursued, i.e., that performing a simulation is accessible to any kind of decision-maker, regardless of the technical expertise.

Unfortunately, currently this is far from being a reality. The main issue is that BIM and BEM define the building quite differently: a BIM model consists of a hierarchy of physical objects with solid geometries, whereas a BEM model is a mathematical description of zones, bidimensional surfaces and adjacencies. Usually, geometry transformations cannot be done without some rework. In addition, not all BEM tools have the same internal representation. This is a long-studied issue in the literature [15,16].

Several approaches to address BIM to BEM workflows have been proposed, mostly conducted by buildingSMART, developing Information Delivery Manuals (IDM) [17], but they rely on expert users' involvement in data modelling and exchange. In parallel, Model View Definitions (MVD) [18,19] have been proposed to help software vendors to support imports and exports through IFC [20], but they have no real support. A recent study by buildingSMART [21] clearly shows that this is still an issue.

The objective of current work is to propose a methodology for solving the data exchange between BIM and BEM tools, mainly focused on the geometrical part, and yield an output file that can be simulated with no errors. In contrast to similar approaches, the main characteristics pursued are:

- Perform the whole transformation automatically in a “single click” process, rather than through a workflow which implies using various support tools and potentially the involvement of different actors.
- Consider the surroundings where the model is located (e.g., neighbouring buildings which act as shadows).
- Don't impose any technical background knowledge in IFC or energy modelling, so that it is affordable to any kind of stakeholder.

In any case, it must be highlighted that addressing the accuracy of the energy simulation outputs is out of the scope of present research, as those do not depend on the model geometry, but on the assumptions taken inside the BEM tool in relation to usage patterns, building control settings and other design hypotheses.

This methodology could have a direct application in early decision stages of a renovation project because:

- There is a vast number of buildings to be renovated in the following decades.
- The information about the building context and surroundings is crucial for renovation and in many cases, freely available through open data.
- In early stages the energy simulation does not need to be so detailed, and involvement of energy modelling experts is not necessary. Thus, we can envisage a single-step workflow affordable for decision-makers or building auditors, with low user interaction, as part of a quick estimation process.
- Because of the previous point, the algorithms should be robust enough to work with a variety of input IFC models from different modelling and data capturing techniques, without involving the use of additional tools in the workflow for enhancing, adapting or postprocessing the BIM model. This challenge also justifies the need of current research.
- This approach also demands the use of open formats to make it vendor independent.

1.2. Literature review

The concepts behind BIM to BEM transformation, along with their main issues and challenges, have been thoroughly addressed by many studies [22–26]. Several of the proposed solutions deal with a user-oriented approach, by defining workflows, data exchanges and design recommendations delivered in the form of IDM/MVD standards, as previously shown. This approach relies on user expertise and in many cases the use of certain software tools (in many cases commercial and subject to licenses).

Other scientific articles, much more relevant for the objectives targeted herein, focus on developing algorithms for computational geometry processing and transformation, which aim to automate part of the user processes and solve some of the gaps that current tools present. Specifically, some BIM tools support the concept of space boundary (SB) as an equivalent to the BEM thermal surfaces and several studies have been carried out to make use of these SBs, to interpret and postprocess them, to fix them or to generate them from scratch.

Table 1 shows a comparison of scientific publications, focused on geometry transformations, according to some of the most relevant aspects tackled in this research, as follows:

- Modelling requirements: it indicates if the proposed algorithm or solution requires generating the model in a particular way or following some methodology or naming using a given software (e.g., Revit [27], SketchUp [28]) or if they are implemented as plug-ins of other tools.
- Spaces needed: it indicates the geometries of physical spaces must be explicitly provided (Y) or can be inferred from the bounding elements (N).
- Clashes allowed: if clashes (overlapping geometries) as well as other inconsistencies (not touching adjacent elements) can be assumed or not.
- Multi-building: if the solution works for multi-building complexes or even districts.
- SBs: if the solution generates a space boundary or similar representation (e.g., surfaces mapped to bounding building elements and identifying adjacent spaces in the front and back side) compatible with the inputs of most widely used energy performance simulation tools, e.g., Energy Plus [29] (one of the most popular open-source energy simulation engine) and gbXML [30] (the most widespread open format for data exchange between BEM tools).

Table 1
Comparison of similar works.

Reference	Modelling reqs.	Spaces needed	Clashes allowed	Multi-building	SBs	Tilted surfaces
Rose & Bazjanac [31]	n/a	Y	N	N	Y	N
Yang et al. [32]	Y	N	Limited	N	Y	N
Mediavilla et al. [33]	Y	Y	Limited	Limited	Y	Limited
Lilis et al. [34]	Y	Y	N	N	Y	N
El-Diraby et al. [35]	n/a	Y	N	N	Y	N
Ladenhauf et al. [36]	n/a	Y	N	N	Y	N
Jeong and Son [37]	Y	Y	n/a	N	n/a	N
Ying and Lee [38]	Y	Y	Limited	N	Y	N
Van Treeck & Rank [39]	n/a	N	n/a	N	N	N
Chen et al. [40]	Y	N	N	N	n/a	Limited

- Tilted surfaces: if it can handle tilted surfaces such as non-flat roofs (quite common case) or non-vertical walls (very rare case).

1.3. Rationale of the research beyond the state of the art

Summarizing, the analysed literature in this topic highlights that:

- The transformation from BIM to BEM is an algorithmically complex issue, prone to errors, which is in some cases solved by imposing some conditions to the input IFC geometry, being the most common:
 - The need to have spaces physically modelled and in most cases perfectly adjacent (no overlaps nor gaps) [31,33–38].
 - Clashing between building elements (e.g., overlapping or unconnected walls or floors) is not usually accepted, only some of them do, with limited approach [32,33,38].
 - For solving the above, some works suggest the use of auxiliary third-party tools (such as Solibri [41]) to audit and fix the model for inconsistencies.
 - Other restrictions as not support tilted surfaces or only certain types of geometries (explicit, non-parametric polygonal representations).
- Some researches (e.g., [39]) propose a way to extract a volume model from a BIM with even no spaces, but it is focused on CFD (Computational Fluid Dynamics) tools as an output and does not cover the generation of space boundaries and adjacent surfaces between spaces in the semantics required by gbXML or Energy Plus.
- With respect to input BIM models, all the test cases conducted are based on models coming from design tools (mostly Revit). None of them show examples where the methodology is applied to models coming from reality capture techniques (e.g., from IFC models originated from a capture through laser scanning or mobile devices). For renovation and energy audit use-cases, these inputs are becoming more and more popular, and they usually produce BIM models which require cleaning and postprocessing for practical purposes.
- In some cases dependency of specific commercial tools is stated, either as authoring tool, or as an environment to implement the algorithm as a plugin, e.g., Autodesk Revit [32,33].
- Multi-building approach is not targeted. It is only mentioned in Mediavilla et al. [33]. However, it requires to manually indicate the external building in the authoring tool and does not implement performance optimisation techniques for models with a very detailed surrounding. The methodology presented herein, first calculates all the bounding boxes of each external object and then implements a visibility analysis for selecting only the relevant surfaces for simulation, which can reduce drastically the number of surfaces and consequently the simulation time.

- Most of them seem still a proof-of-concept with no performance tests acknowledged. Only [38 and 40] show promising performance results (few seconds for big models). [31] also provides figures, but for medium-sized buildings (around 50 spaces) requires a running time of several minutes.

After this analysis, it can be concluded that none of the similar works cover the whole process of (1) geometry pre-processing and cleaning (2) space generation (3) boundaries detection and creation (4) external building influences analysis (5) consistency checking and finally (6) to export the model to open BEM formats (e.g., gbXML or Energy Plus input files). All of them leave some of the steps uncovered or assumed as an input requirement. Moreover, they do not offer results for potentially more complicated models coming from scanning techniques. Thus, a holistic approach that can cover all these aspects altogether seems justified.

The approaches to generate an analytical model in literature can be classified in two main categories: one is to identify all real surfaces that bound the spaces and their enclosing objects (walls, slabs...) and to match them and detect adjacencies and intersections. A second one (which is the selected one for the presented methodology) is the opposite: generate a simplified building topology of nodes and edges (i.e., a graph) and use them to reconstruct the space volumes and connection boundaries from there.

In graph theory [42] (a branch of mathematics), a graph is a collection of objects (called nodes or vertices), which are pairwise related. These relationships are denoted by edges. When the relationship follows a particular order or direction the graph is directed. Nodes and edges can represent any kind of concept and relationships, being a foundation for many fields, such as the semantic web, neural networks, and many other applications. In the context of present work, graphs will be used to represent geometrical and topological relationships between building elements, to enable the automated identification of adjacencies, minimal routes, thermal surfaces, and their function (external, internal, ground, etc.).

The reason for using graph techniques is that it is less affected for the real BIM geometry and thus more robust when generating correct topologies. Moreover, tests conducted in this work also proved that it is very efficient in time.

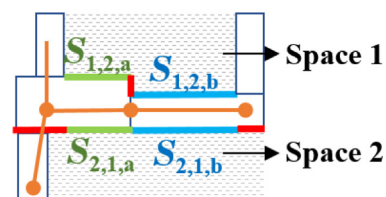


Fig. 1. Graph vs physical surfaces.

This is illustrated in Fig. 1. The first approach consists of processing the spaces geometry (shaded areas) and their bounding surfaces and finding the counterpart space and the in-between building element surface. Thus, adjacencies depicted in green and blue will be correctly matched in most cases, but the ones in red will be problematic. This will be very common when we have walls of different thicknesses and irregular geometries. On the contrary, by creating first a graph (orange dots and lines) and reconstructing the surfaces from there is a more robust approach. Nevertheless, two important aspects must be considered:

- The node generation process must be efficient enough to detect duplicate or redundant nodes and fix those in an appropriate manner. This is one of the main challenges, which involves an optimisation of the internal configuration tolerances used.
- The volumes derived from here will be bigger than the real physical space volumes, which can lead to deviations in the results of the energy simulation.

The proposed methodology also considers how to deal with these two issues.

Some works already explore graph techniques. However, in some cases they are focused on volume detection (e.g. for CFD simulations) and the proper identification of space boundaries is not developed [39] or they require properly modelled spaces and non-overlapping geometries as input [31]. Yang et al. [32] do not use this term, but in practice they also propose a graph generation of edges and loops from wall geometries. However, they do not implement a pre-processing step to clean up input complex geometries (even parametric representations and tilted rooftops), neither a post-processing step at the end to automatically fix potential errors and clashes in the generated surfaces, which is covered herein.

The optimisation of the multi-building approach is addressed by implementing a visibility analysis for filtering relevant surfaces for simulation (including also self-shadowing effects), which can reduce drastically the simulation time, as it has been demonstrated with some test cases. This is a crucial aspect in urban environments and there is a growing need to consider the building in its context [43,44], which is also an important result achieved.

Finally, for the sake of universality and interoperability the use of open standards is mandatory, thus IFC has been used as input source and gbXML and IDF (Energy Plus) as output formats.

The proposed methodology has been applied to two test cases. These models have also been tested with the gbXML export option in Revit (section 4.2), being the results less satisfactory (quite incomplete or directly not valid), which demonstrates the potential of the proposed methodology.

The results are very promising. First, the procedure can be universally applicable regardless on how the IFC is created, having in mind especially its use in renovation scenarios (we must bear in mind that approximately 60 % of current building stocks are likely to remain in use by 2050 in the European Union, United States, and Russia [45]). This was a key motivation for the work since renovation and energy performance audit scenarios do not usually have a BIM and we cannot always rely on specialised desktop authoring tools for preparing and correcting the model. The most novel approaches go in the direction of creating the model automatically via on-site capture techniques or computer-vision based floorplan processing. Since IFC models are not created by skilled designers, but by algorithms, they are prone to have more geometry clashes and inconsistencies. To include these algorithms in the workflow would allow any energy auditor to assess the building without modelling knowledge or the use of third-party tools. One of the IFC files tested in section 4 was created with one Scan-2-BIM based solution, ARtoBuild by Bimeo [46].

The rest of the work develops and tests the methodology as follows: section 2 summarises the methodology, which is detailed step by step in section 3. Then, section 4 shows its application to some test buildings whereas section 5 presents the conclusions, limitations, and outlines the future research directions.

2. Summary of the methodology

The target of the present work is to generate a building energy model from a multi-scale BIM model, i.e., where multiple building instances can coexist together with detailed internal decomposition (storeys, walls, spaces, etc.) of one or several of those buildings. For this purpose, graph techniques are used.

2.1. Input model requirements

As it has been already introduced, the methodology has two main goals: (1) to drastically reduce the requirements of the input IFC in terms of geometrical quality (e.g., clean, not clashing geometries) or semantics (proper modelling of `IfcSpace` entities), making it applicable to any kind of IFC, regardless of its generator tool or user and (2) to be efficiently applicable to multi-scale models.

In relation to the building structure, building elements that form the thermal envelope of each zone must be represented with the adequate IFC class, namely: `IfcWall` (or any of its subclasses), `IfcCurtainWall`, `IfcSlab`, `IfcRoof`, `IfcCovering`, `IfcDoor` and `IfcWindow`. This is a soft requirement, met by any BIM authoring tool. Each envelope element must have a geometry or be decomposed in parts with their own geometry (e.g., layered walls).

In relation to the multi-scale approach, when the model represents a multi-building complex it must follow one of the three approaches depicted in Fig. 2. Option 1 is the most common approach followed by BIM authoring tools, because they are focused on single-building design, thus, when exporting to IFC they only export a single `IfcBuilding` entity for a project, whereas external elements to the building (such as other buildings or landscape elements) are represented as generic items (typically `IfcBuildingElementProxy`) instead of `IfcBuilding`. These external items are typically assigned to the target buildings' ground floor. In any case, the IFC schema supports the assignment of several instances of `IfcBuilding` to one `IfcSite` (Option 2), some of which could be further decomposed in their internal structure in a hierarchical way. We could even have a project with multiple sites (Option 3), although it is extremely unusual.

2.2. Sequential steps of the methodology

The procedure described in this work takes as an input an IFC model and generates an analytical model for each of the `IfcBuilding` instances found. In each iteration one `IfcBuilding` is considered as target and the rest as external buildings. The effect of external buildings will mainly be related to shadowing effects and the algorithm is not affected by how these external buildings are defined (as `IfcBuilding` or as `IfcBuildingElementProxy`). However, only buildings with internal structure are valid target building candidates, i.e., `IfcBuilding` instances modelled as a single empty block will never be processed (only as obstacles for other buildings).

The creation of the analytical model for each building (each iteration of the multi-building model) is a complex procedure which is split in a series of sequential steps, summarised in Fig. 3 and described in detail in subsequent chapters.

- Step 1: in each storey create a graph by detecting and processing all nodes, edges and loops.

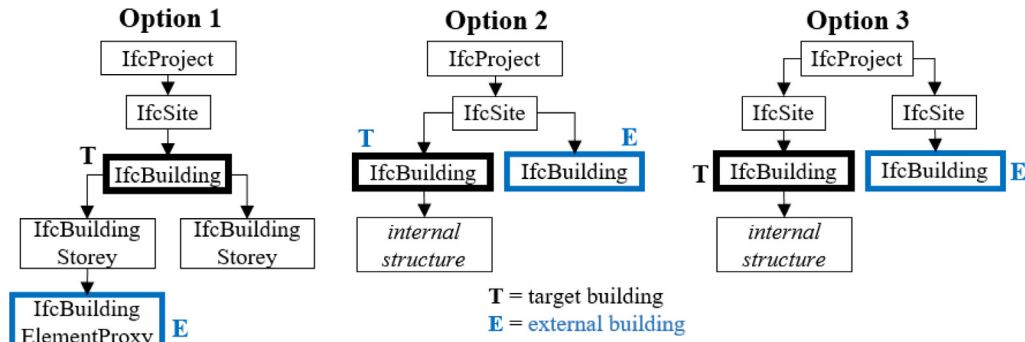


Fig. 2. Options for multi-building modelling in IFC.

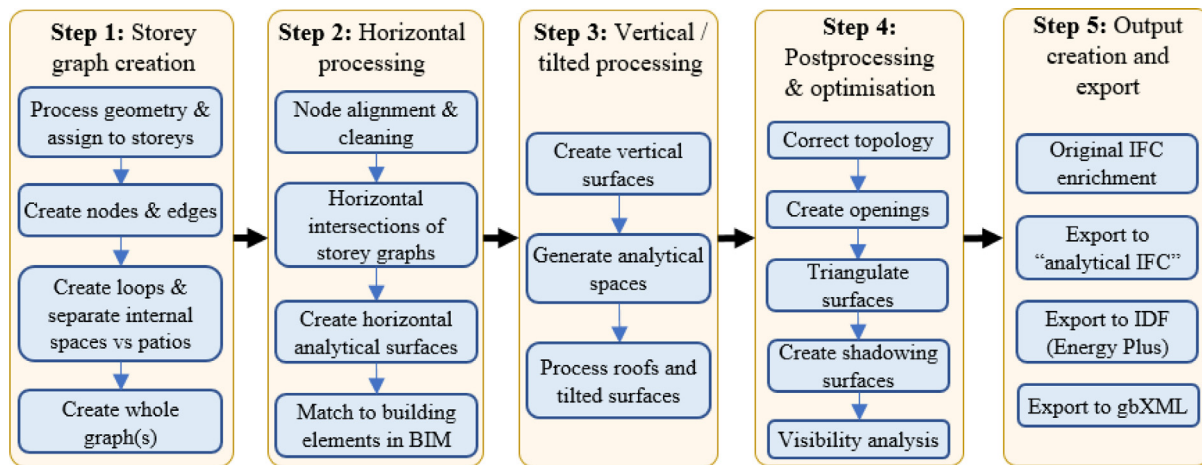


Fig. 3. Process summary.

- Step 2: perform boolean operations between each pair of adjacent storeys and create the corresponding analytical surfaces (typically matching slabs in IFC).
- Step 3: process vertical and tilted surfaces (the latter ones using clipping operations) and generate the analytical spaces.
- Step 4: correct any error found (e.g., overlapping or self-intersecting surfaces) and generate the openings. Then, create the external shadowing surfaces and optimise the model by removing non-visible ones.
- Step 5: enrich the original IFC model and export to various formats, as IDF (Energy Plus) or gbXML.

- Create a convex hull of the polygon. Here Graham scan method has been implemented [47], which takes a polygon like the one in (b) as input and generates the orange contour in (c).
- Apply the rotating callipers technique [48,49], which consists of drawing all possible rectangles which are collinear to a polygon segment and enclose the polygon completely, as in Fig. 4(d). Applying to the result from (c) we obtain the green rectangle in (e).
- Extrapolate to 3D. For all possible polygons in the object boundary a 3D box is built by considering this polygon as the floor ($z = 0$ plane). In Fig. 4, (f) two possible iterations of the same tilted object are shown, being the lower the one that yields the least volume. In (g) a not valid iteration is depicted, since the base polygon selected has a non-convex vertex, thus part of the object is outside the box (below $z = 0$, in red). That is the reason for calculating the convex hull in (c).

3. Detailed description of the methodology

3.1. Step 1: Graph creation for each building storey

The initial step is to process the geometries of the relevant products for the analytical model (walls, slabs and coverings, roofs, doors and windows) and simplify them before creating the graph. The sequence is as follows:

- Process each object's geometry. All types of IFC geometries are supported (boundary representations, extrusions, clippings...), including the decomposition of an object in parts.
- Generate the minimal bounding box (the enclosing 3D box with the minimum possible volume) for speeding up next calculations, as in Fig. 4 (a). It is a 3D extrapolation of the minimum bounding rectangle concept in the 2D plane, which requires to:

Creation of nodes, edges and loops

The key physical element for the graph creation for each storey is the wall since it constitutes both the internal partitions and the boundary with the external environment. Subclasses of IfcWall and IfcCurtainWall are considered, including possible internal decomposition.

Nodes are calculated as the points where a pair of wall segments intersect. This is done by computing the central axis of each wall. Walls' intersections can come in a variety of forms depending on the modelling technique. Thus, we could have overlapping walls (Fig. 5(a), walls 2 and 3), non-rectangular connections (Fig. 5(a),

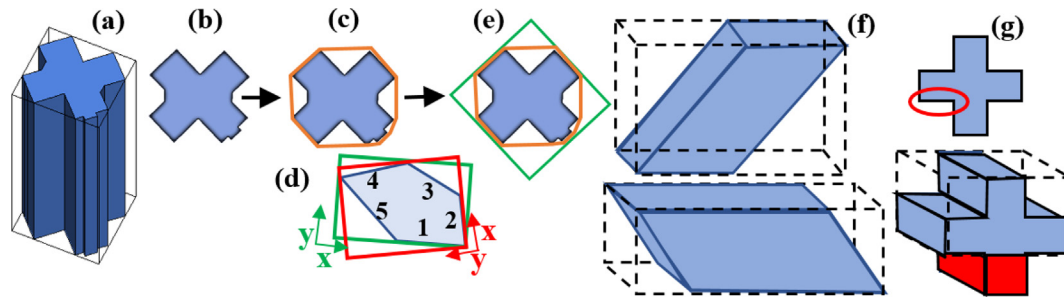


Fig. 4. Creation of bounding boxes.

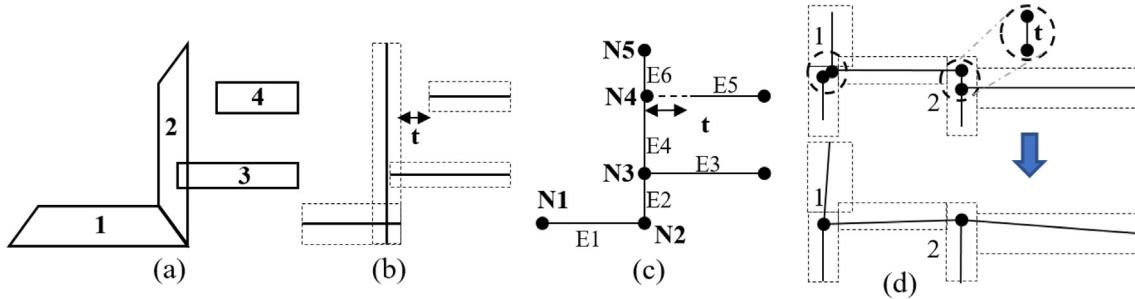


Fig. 5. Creation of nodes and edges from a set of walls.

walls 1 and 2), or non-touching geometries (walls 2 and 4), having a tolerance distance t , as in Fig. 5(b). To make the process homogeneous, bounding rectangles are calculated for each wall projection (Fig. 5(b), dashed lines), being the axis the central line between the two longest sides. Then, by computing the intersections, the whole set of nodes is obtained, labelled as N1, N2, etc. in Fig. 5(c).

As mentioned before, there can be situations where walls do not completely touch, having a tolerance distance t . In these cases, a threshold value (t_{max}) has been defined for t . When $t \leq t_{max}$, a real node is generated, as in Fig. 5(c) for node N4. When $t > t_{max}$ no connection is produced (open space). The application of the methodology to selected test cases (section 4) has helped in fine-tuning the values of these tolerances.

There could also be a variety of walls geometries and thicknesses, resulting in two or even more nodes for the same logical connections, as shown in Fig. 5(d), dashed regions 1 and 2. When this happens, a tolerance value t has also been defined to merge the nodes that are closer than this distance. These adjustments could provoke that the angle between the edges is slightly distorted, but for analytical purposes, i.e., for launching a simulation, it has a neglectable impact.

Fig. 6 shows an example run using the ground storey of a test IFC model. On the left side the physical building elements are represented in a footprint view, where each wall is denoted by W1, W2 and so on. On the right side the following concepts are represented:

- Created nodes (N1, N2, etc.) with a naming convention for related walls and the connection type. Thus, in “N11-W2_E-W5_S” node 11 is the point where the end (E) of W2 and the start (S) of W5 meet. N3-4-10 means that nodes N3, N4 and N10 have been merged into one, according to the criteria shown before.
- Some walls also have middle (M) connections apart from start/end (the case of W5, W7, W13, W14).
- We could have walls with unconnected ends (e.g., W7 in the left image). In these situations, a threshold distance (d_t) is defined,

so that when the distance from this end to the closest detected node is higher (as with $d_1 > d_t$) this segment is considered relevant and consequently a node is created, shown as N26 in the right image. Since it is an unconnected end, only one wall (W7) is assigned to the node, where subscript N means that the node represents an unconnected end. On the contrary, $d_2 < d_t$, thus, this end is filtered and node N27 is not created. In turn, N13 is considered as the starting node for the wall (labelled as W7_S). In a second run, unconnected ends will be tagged as start or end, depending on the case (as it happens with N26 in Fig. 6).

- Finally, an edge is formed by exactly-two neighbouring nodes. In the right image of Fig. 6, each segment represents a different edge). Thus, there are walls which produce a single edge (W1, W2 or W3, etc.), two edges (W13, W14, etc.) or even three (W5).

Next, all possible closed loops must be detected, which will eventually be interpreted as space geometries. A closed loop is defined as a sequence of nodes connected through edges, where the last node in the sequence is also connected to the first one by an edge. This is in turn a complex process which is also divided in partial steps, being the first one a branching exploration of the nodes until all possible sequences, starting from a given node, have been identified. This is illustrated in Fig. 7. In the left side, the previously created nodes and edges are depicted (for simplicity, nodes are represented by numbers, not by names N1, N2, etc.).

The process, starting at a randomly chosen node (e.g., node 1 in iteration #1), is summarized as follows:

- Identify all nodes connected to this node and create a temporary loop for each (iteration #2). In this case, two branches are created (nodes 2 and 3). Each open branch is represented by a green circle).
- For each active branch, repeat the branching process starting from the last node visited, including backward routes (iteration #3). Every time a previously visited node is reached, the branch is closed (red circles in Fig. 7) and not further explored.

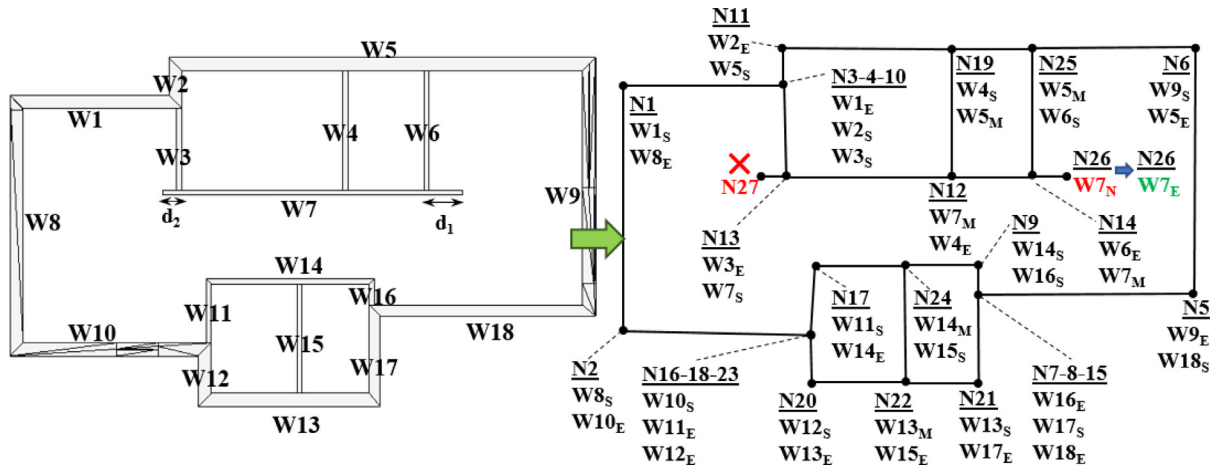


Fig. 6. Creation of nodes and edges from a sample IFC building storey.

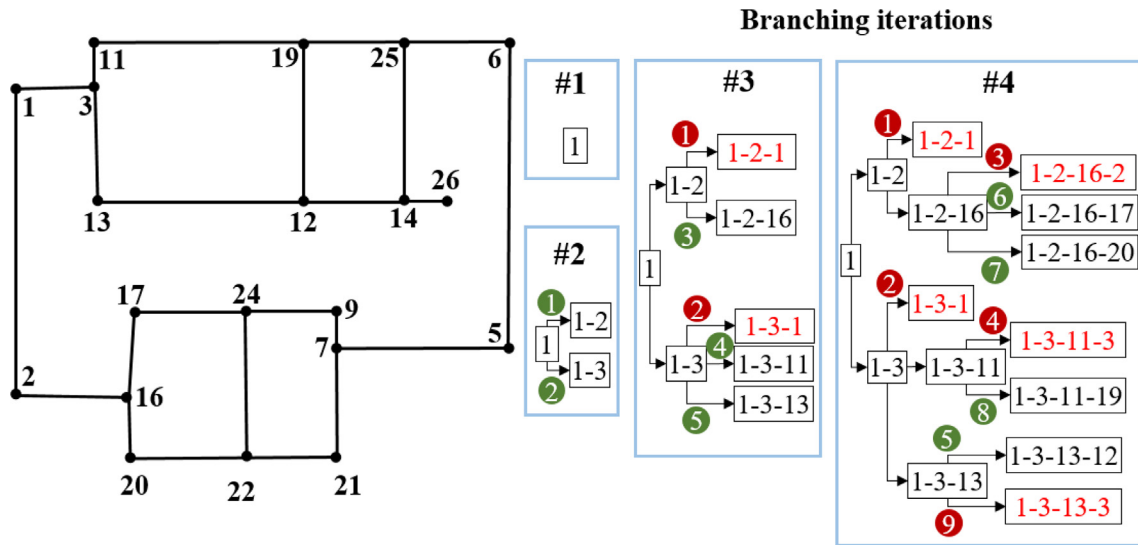


Fig. 7. Simulation of some iterations of the loop detection process.

- In each iteration, the process of opening and closing branches will coexist, until eventually all branches are closed (all nodes have been visited, even following different paths). When all branches are closed the exploration process stops. Fig. 7 shows a simulation until the fourth iteration.

Next, a cleaning process is run to leave only the valid branches for creating loops. The process consists of:

- Remove the branches not forming polygons. This happens when an edge is repeated, i.e., when between two visits to a given node there is only one single node (e.g., 1-2-1). According to this criterion, none of the 5 branches in red in Fig. 7 (iteration #4) forms a valid loop.
- With the remaining branches, the ends are trimmed until they form a valid closed polygon. The trimming consists of detecting the repeated node and removing all the nodes which are not between the two appearances of this node. For example, in Fig. 8(a), it can be noticed that in the first iteration the two loops visit twice node 3 and everything outside this sequence is removed (in this case, only node 1).

- Once this is done, all possible permutations of a given loop are identified and removed, since they represent the same topological boundary (1-2-3-4 is the same as 2-3-4-1 or 4-3-2-1). This is shown in iteration #2 of Fig. 8(a). After this process, all the remaining loops will be unique.
- Previous steps perform a comprehensive exploration, detecting all possible closed loops. However, since the objective is the detection of space boundaries, only atomic loops (not having smaller loops inside) must be considered. For instance, in Fig. 8(b), loop L3 (the closed sequence 3-13-12-14-25-19-11-3) can be split into two smaller loops (L1 and L2), being L3 just their union.

The procedure described, follows the implements deep first search (DFS) algorithm, where the complexity of the computation time is usually $O(|N|+|E|)$, being $|N|$ and $|E|$ the number of nodes and edges, respectively, with these particularities:

- DFS is usually applied to IA optimisation problems, shortest path finding or game-solving (e.g., chess). In those cases, the number of nodes can be enormous in comparison to the number of walls intersections in a building.

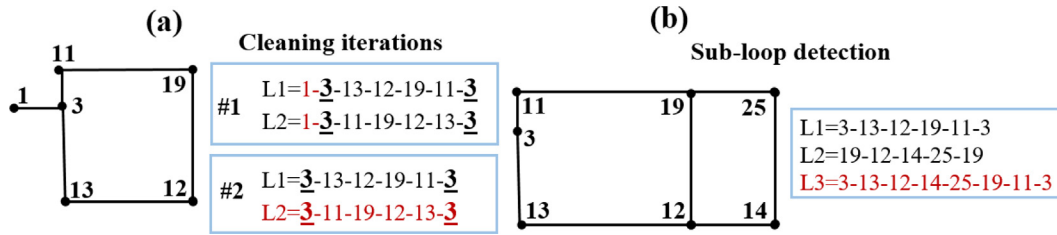


Fig. 8. Cleaning process for the loops (in red, discarded loops).

- The branching factor (i.e., the number of non-visited neighbouring nodes from a given node) is quite small (2–3) in comparison to mentioned applications (e.g., in chess-solving the average branching factor is 35). Additionally, many of these branches end-up rapidly in visited nodes, so the exploration ends.
- Thus, although the temporary number of open branches can be big the process rapidly ends after few iterations (e.g., one of the test cases shown in section 4.1 took less than a minute for a 4-storey building with 66 wall segments).

In the end of the process, the storey is completely covered by loops (e.g., L_1 to L_9 in Fig. 9(a)), with no blank spaces, being the storey boundary the union of all atomic loops (L_S). Any other loop, such as L_A and L_B shown in (b) is not valid. The condition for invalidating a loop L_i is:

- At least one other loop L_j exists which is completely inside this loop ($L_j \subset L_i$).
- There is at least one edge (and consequently two nodes) shared between L_i and L_j . This implies that L_C cannot be discarded in Fig. 9(c), because L_D could represent a patio or any kind of internal space, which acts as a voiding element for of L_C . This is checked by vertical raytracing from the space centroid, as seen in (d). If the ray crosses at some point any building element geometry in its way to infinite z, then it is a real space. Otherwise, it is an external open area. Only loops not sharing any edge with the external boundary can be candidate patio loops, i.e., in the case of (a) only L_5 .

Applying these rules to the original example illustrated in Fig. 7, the output will be a set of 5 atomic loops represented in Fig. 9(e). Unconnected nodes such as N26 in Fig. 7 have no influence at all for loop creation, and the corresponding edge will be an internal edge within the space.

In some complex scenarios there could be more than one graph associated to a single building storey, e.g., when a building is composed by different parts with variable heights, like in Fig. 9(f). A horizontal slice in storey S_k results in two distinct graphs $G_{k,1}$

and $G_{k,2}$ since no node from graph G_1 can be linked to any node of graph G_3 through an edge. The procedure described in current work supports this generic situation, although all the tests performed herein have only been done using IFC models which contain a single graph per building storey.

3.2. Step 2: Horizontal processing: Combine analytical storeys.

The previously created storey-level graphs will be combined into a single analytical model, by processing the intersections between consecutive storeys. Let us assume the following convention (Fig. 10, left):

- A building is decomposed in n ordered storeys from bottom to top, S_0, S_1, \dots, S_{n-1} . Each of them has m analytical graphs $G_{i,j}$. Storeys with two or more disjoint graphs are applicable in the methodology, but this situation has not been tested yet.
- Each iteration of the procedure described herein consists on comparing a given graph in storey i ($G_{i,c}$), with one graph in storey i-1 ($G_{i-1,p}$) and another one in i + 1 ($G_{i+1,n}$), where c = current, p = previous and n = next.
- The intersections have been created via boolean operation. The typical notation used in set theory (Fig. 10, right) will be used henceforward.

Creation of analytical surfaces by storey intersection processing

Analytical surfaces are planar surfaces taking part in heat transfer processes. All spaces must be completely bounded by non-overlapping analytical surfaces. There are many standards which follow this approach, such as gbXML [30], a BEM standard supported by many tools, or the IfcRelSpaceBoundary [50] entity in IFC, implemented in different formats. Considering that, created surfaces have the following characteristics:

- All surfaces have a function: internal (in contact with another internal surface), external (exposed to sun or wind), ground (in contact with the ground) or adjoining (in contact with another building).

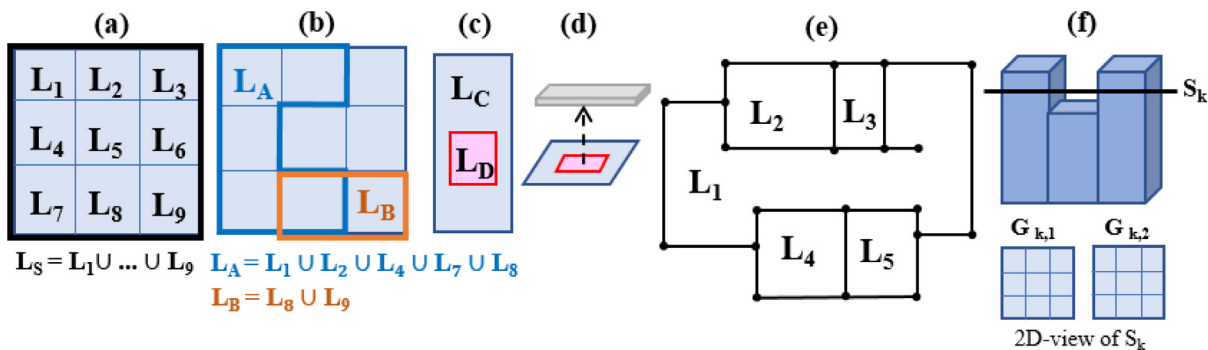


Fig. 9. Valid and not valid loops within a storey.

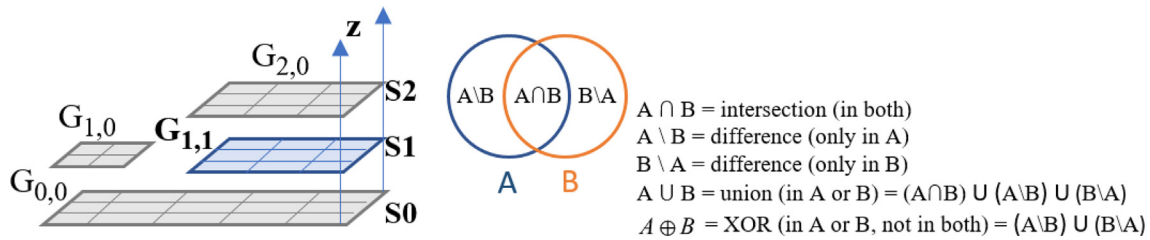


Fig. 10. Graphs notation (left) and node alignment (right).

- Additionally, an internal surface can only be in contact with exactly-one surface. If one surface is in contact with two different surfaces it must be split in two, to ensure a 1 to 1 correspondence.
- All surfaces have a type, taken from a closed list of options: wall, floor, ceiling, roof, door, window or virtual (not corresponding to a real physical building element). There are two additional types considered for surfaces not belonging to any space, but relevant for shadowing or obstacles: self-shadow (such as a salient or overhang) and external shadow (e.g., a vertical surface representing a nearby building).

Considering this, the creation of analytical surfaces is split in smaller steps. In many of them it is necessary to perform boolean operations with polygons (union, intersection, difference, etc.), which can be done with various geometry processing libraries. For current analysis Polybool-Java was used [51], which is a Java implementation of an existing algorithm proposed by Martinez et al. [52].

- 1) In the case of the lowermost storey, each loop yields an analytical surface which represents the contact with the ground (see Fig. 11 (a)).
- 2) For any other storey, all possible intersections between each loop L_c in the current graph ($G_{i,c}$) with each loop L_p in the previous storey graph ($G_{j,p}$) are processed, by projecting the corresponding polygons in the same plane. This is denoted by $L_c \cap L_p$ in Fig. 11. Four situations can occur:
 - i. If both loops represent internal spaces, the intersection produces two twin internal surfaces: S_1 as seen from L_c (representing a floor) and S_2 as seen from L_p (representing a ceiling), shown in Fig. 11(b).
 - ii. If the L_c represents a patio instead of an internal space as shown in Fig. 11(c) by means of a not-shaded area, only one analytical surface will be produced: an external ceiling as seen from the loop below. Likewise, if L_p represents a

patio as in Fig. 11(d), the intersection represents an external floor (i.e., a salient floor in air contact) from the perspective of the L_c .

iii. Finally, when both loops are patios, no surface is produced, and the situation is discarded.

- 3) Once all the intersections with all other loops have been detected, the remaining surface must be processed, which, by definition, must be external. A quick way to identify this is by computing the difference with the loop which represents the boundary of the previous storey (except when we are computing the lowermost storey) and the same with the next (except when we are computing the topmost storey). This is shown in Fig. 12(a) and (b). In the first case, the orange-shaded area represents the previous storey, thus the difference $L_c \setminus L_{sp}$ can be considered as a floor. In the second case the difference represents the next storey, thus $L_c \setminus L_{sn}$ is a ceiling. It must be noted that in the unlikely case that we have a storey with more than one graph, as in Fig. 12(c), both polygons must be treated as a single multi-polygon when computing the difference, thus this example would not yield any result. To assign the function to the surface, the z coordinate of any random point is checked. If z greater than 0, which will happen in most situations, the surface is in fact external, otherwise it is a surface below ground (also external, but the meaning of “external” in this thermal context is “in contact with outdoor conditions”).

Assignment of horizontal analytical surfaces to building elements

For thermal analysis purposes it is crucial to assign the surface to the physical building elements where they are located. Thus, the distance from each previously created surface (taken from the centroid) to all possible building elements is calculated (Fig. 13(a)) and the closest one is detected.

Almost always the surface will match an existing element, thus the closest element will be at zero distance or some neglectable

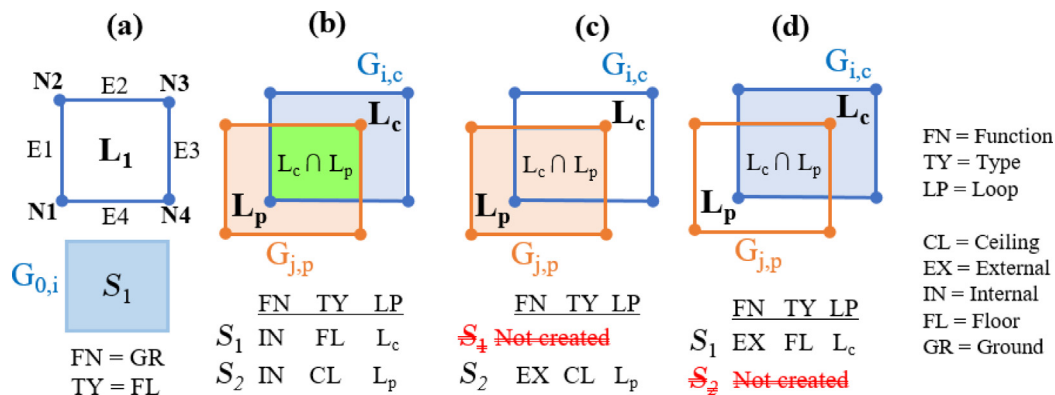


Fig. 11. Intersections between storey loops.

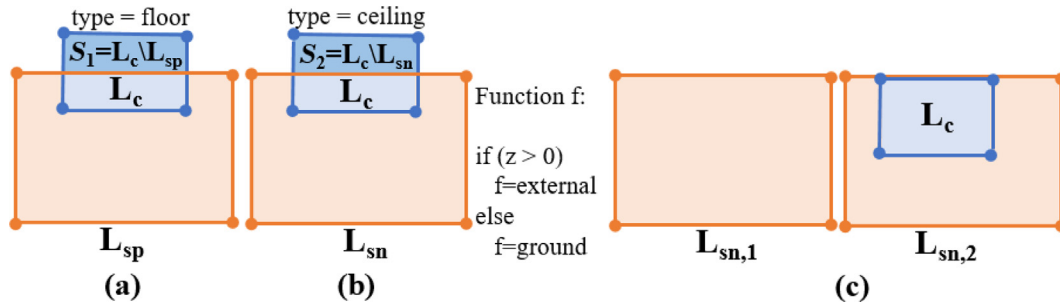


Fig. 12. Intersection with the storey boundary.

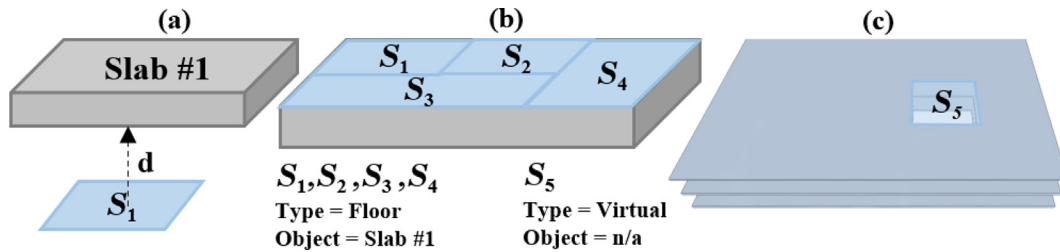


Fig. 13. Assign horizontal analytical surfaces to building elements.

tolerance due to rounding errors, as in Fig. 13(b). However, there could be situations where this is not true, as the example in Fig. 13(c), where the analytical surface matches a hole of an element (e.g., a hole in a slab which represents the staircase or elevator shaft). If so, the surface's type is changed to "virtual".

3.3. Step 3: Processing of vertical and tilted surfaces

Creation of vertical analytical surfaces

After the horizontal processing, the next step is to create the vertical analytical surfaces, which is a simpler process, because most of the complexity was already solved during the loop creation. It consists of extruding the created edges until the next storey level, creating rectangular surfaces. Some considerations to be done:

- When an edge belongs to two loops, such as E_a (part of L_1 and L_2) in Fig. 14 two identical internal surfaces are created, with opposite normal vectors, shown in green and orange in (a). These surfaces are marked as adjacent to each other, which is of utmost relevance for any energy model.

- On the other hand, edges belonging to a single loop, as E_d (only part of loop L_4), will produce a single surface, which will be marked as "external" if $z > 0$ or "in contact with ground" otherwise.
- Additionally, although quite less frequently, there could also be edges belonging to no loop at all, since they are part of an unconnected end, such as the two red edges E_j and E_k shown in Fig. 14(c). A surface is also created in this case but assigning no thermal function and no adjacent surface to it.
- Finally, all the surfaces in storey i are extruded until the elevation of storey $i + 1$. A special case is the last storey. In this case, if any valid loop has been found a temporary set of vertical surfaces is created. Its post-processed will be explained afterwards.

Creation of analytical spaces

After all the horizontal and vertical loop intersections have been processed, everything is ready for the creation of the analytical spaces and their mapping to surfaces, which is the goal of the whole process. Thus, these steps are followed for each loop L_j in the storey graph G_i which represents an internal space (i.e., not a patio):

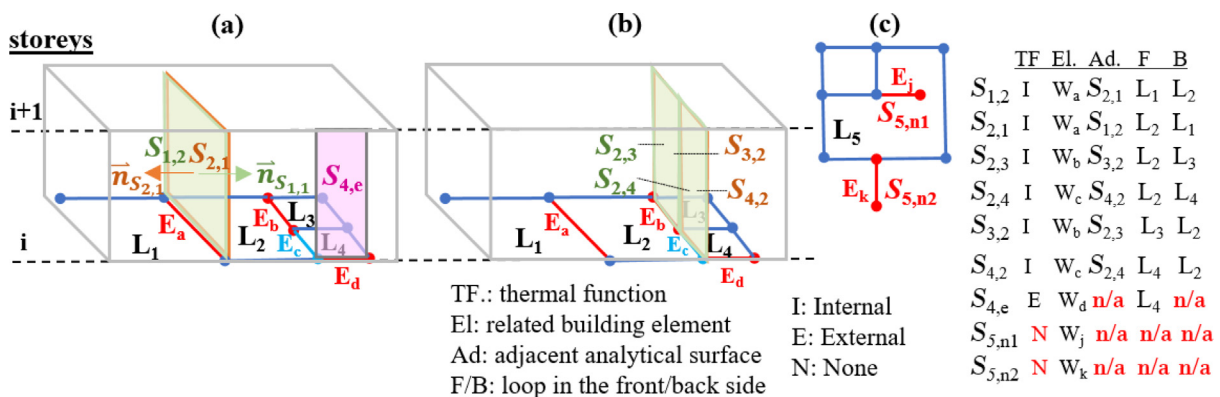


Fig. 14. Creation of vertical analytical surfaces.

- Create the space by extruding the loop polygon until the next level. If the loop belongs to the last (rooftop) storey, an initial arbitrary extrusion distance is assigned, which will be processed later on. An example is shown in Fig. 15 (a), where the space from loop L_2 is represented in blue.
- Assign to the space all the horizontal and vertical analytical surfaces of the corresponding loop. The example in Fig. 15 shows a common situation where the number of surfaces of a space (represented with different colours in (b)) is higher than the number of sides (blue polygons in (a)), because we must account for intersections and adjacencies with other loops/spaces both horizontally and vertically.
- For each analytical surface S_k in the storey replace the front and back loop assignment done in the previous step (and shown in Fig. 14) by the real space they represent. E.g. for the surface $S_{2,1}$ the front object is changed from loop L_2 to Space 2 and the back object from L_1 to Space 1.

Processing of roofs and tilted surfaces

In the described procedure there is still a special case to handle: the various typologies of roof shapes. This will mostly apply to the roof storey (the uppermost one), but in some cases there could be roof instances in other storeys.

The processing done for each of the situations depicted in Fig. 16 is listed herein:

- Roof storey with flat roof and walls. This configuration does not produce any graph. In this case, each horizontal surface of the previous storey (i.e., the footprints of each analytical space) is projected into the roof plane, creating new analytical surfaces of type = “roof” and function = “external”.
- Roof storey with flat roof but wall instances found forming a loop (L_r in Fig. 17 (a)). In this case, the previous steps have created temporary analytical surfaces and spaces for the rooftop storey by intersecting L_r with the previous storey ($S_{r,a}$, etc.). At this point, the algorithm checks the distance d from each surface to the closest building element above to see if extra spaces must be created. Since we have a flat roof d is neglectable (in Fig. 17(a) it is exaggerated for visual purposes) and no extra spaces are created. Thus, the previously created surfaces are kept, as roof surfaces of the storey below (as seen in the top view).
- In this case, unlike the previous one, non-flat roof surfaces exist, thus the previous check will yield a non-neglectable value for distance d , as seen in Fig. 17(b). Then, the closest plane polygon for all roof surfaces is processed, which act as clipping polygons for the temporary surface and space geometries created in previous steps. The result is shown in Fig. 17(c), where each final surface has a different colour. In this case, the surfaces are not rectangular, but can have more complex shapes and orientations.
- A similar case to the one in (c), but also with clipped walls.

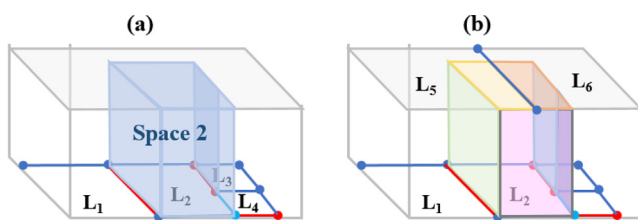


Fig. 15. Creation of analytical spaces.

- A more complex building where there are tilted roof surfaces, not only in the last storey, but in in-between storeys too. This case has not been addressed yet, since it involves also post-processing horizontal adjacencies increasing the level of complexity. However, it is an identified possible future enhancement.

3.4. Step 4: Postprocessing and optimisation

Once all the analytical spaces and their bounding surfaces have been generated, some imperfections created by the algorithms may still exist and must be fixed before creating the openings. Then, in order to optimise the transfer to simulation engines, shadowing surfaces are also auto detected, discarding all those that are not visible from any point in the building. Due to some input requirements in some engines, external surfaces are also triangulated.

Automatic surface topology correction

Almost all simulation engines require that analytical spaces are perfectly closed, i.e., their surface do not self-intersect, nor leave blank spaces. It may seem that the process carried out up to now guarantees this, however many of the steps deal with fixing and aligning nodes and performing various intersections and boolean operations, which could produce errors due to various reasons: simplifications when creating edges and planes, not proper choice of internal tolerances or rounding errors by the CPU, being sometimes cumulative. Even if the errors are not very big (could be just in the order of few mm. or 1–2 cm.), some simulation engines could be very sensitive to this, although some others could self-fix the geometry or ignore some errors. However, for being in the safe side, this procedure includes automatic surface topology correction features.

First, invalid polygons (less than three vertices) are removed from the model.

Secondly, the model is cleaned by removing the polygons which produce noise or are created because of the tolerance values selected throughout this process or due to rounding errors. The narrow yellow polygon shown in Fig. 18 (a) is a good example of this. It is not easy to define the most efficient criterion for filtering these polygons. However, the measure of a polygon compactness seems to be a good approach. In brief, the compactness C can be defined as how close a shape is to a circle by comparing area (A) and perimeter (P) using this formula proposed by Li et al. [53]:

$$C = 2 * \frac{\sqrt{\pi * A}}{P}$$

Some examples are depicted in Fig. 18(b). The compactness of a circle is 1, while polygons resembling a square have bigger values than very elongated ones. An option is to define a threshold value C_t , to filter polygons where $C < C_t$. However, to avoid false positives (e.g., very long corridors) an additional condition is introduced by defining an area threshold A_t , with the same idea of discarding polygons where $C < C_t$ and $A < A_t$ simultaneously. Test cases in the end have helped to tune these limit values.

Finally, the vertices and edges of the surfaces are adjusted to match the space geometry vertices. After this cleaning process is done, it is likely to encounter situations like the ones in Fig. 18 (c), due to rounding errors or internal tolerances of the algorithms. Thus, for each vertex in the surface polygon the closest vertex of the space geometry is found and if the distance is non-zero but smaller than a tolerance value (e.g., the two points inside the red circle in Fig. 18(d)) they are all merged into the same point.

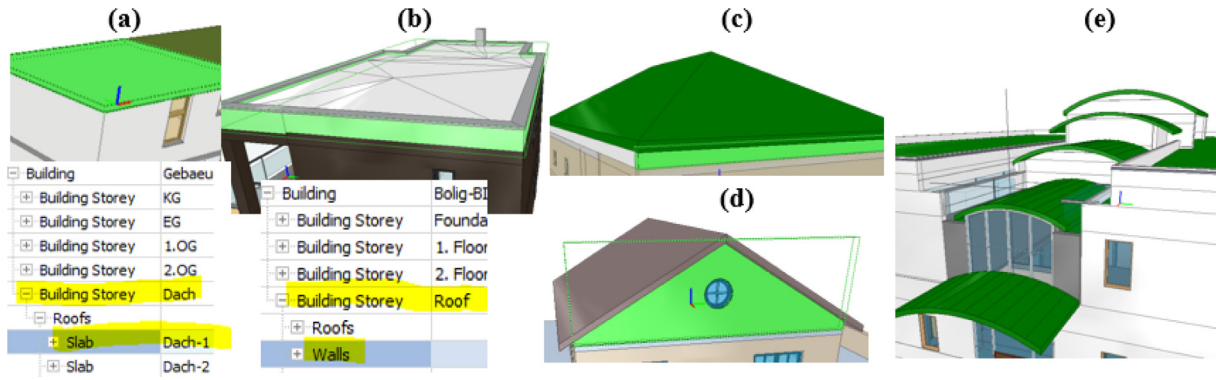


Fig. 16. Different roof configurations in IFC.

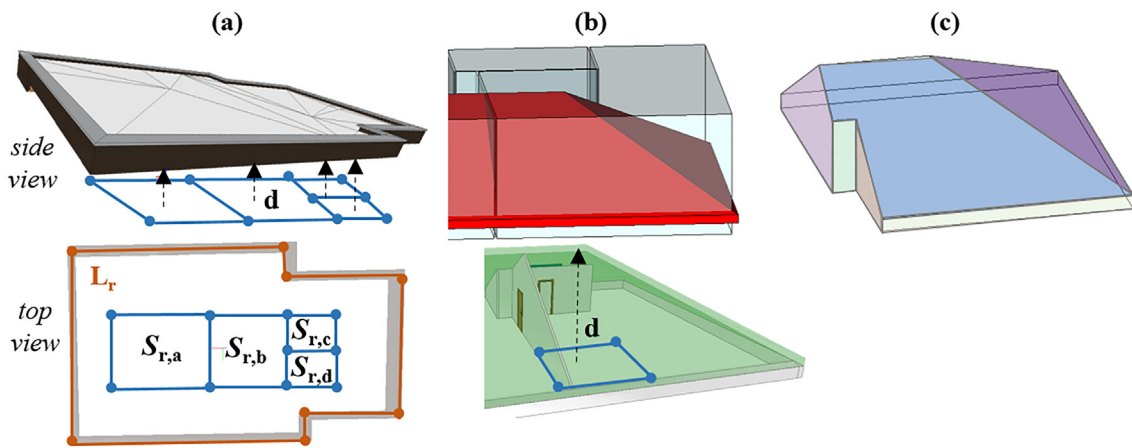


Fig. 17. Generation and/or clipping of roof spaces and surfaces.

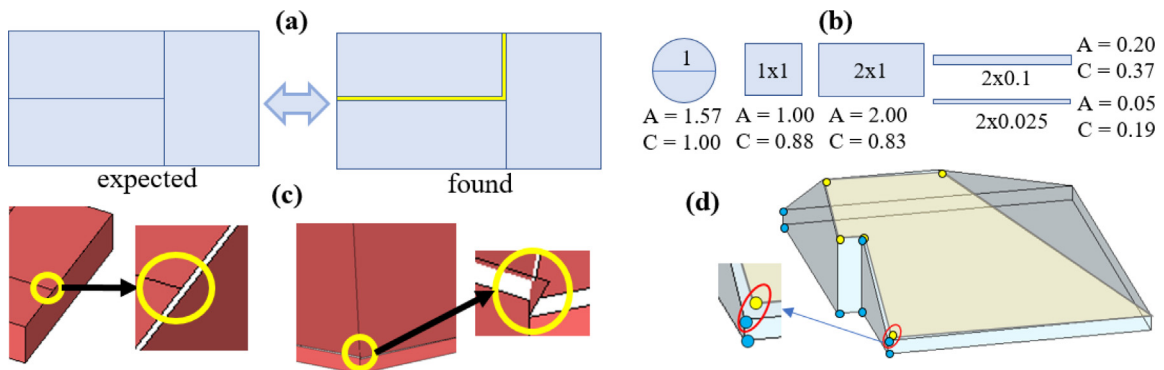


Fig. 18. Surface topology correction.

Creation of openings

Once all the space boundaries are properly created, but still limited to the opaque envelope, everything is ready to create the openings (sub-surfaces for doors or windows). The procedure for each IfcDoor and IfcWindow instance found in each storey is as follows:

- Get the host wall (e.g., W1 in Fig. 19, left), either by parsing the IFC relationships or by pure geometrical processing. Bounding boxes are used instead of the detailed geometry.
- Each wall can split in several analytical surfaces, thus, the closest one is detected, and the bounding box is projected into it (e.g., surface $S_{3,2}$ in Fig. 19, right).

- An opening is created in the analytical surface $S_{i,j}$, named as $S_{i,j,k}$ (opening k in surface i,j). In the previous case, $S_{3,2,1}$, named. If the surface is internal, its adjacent or counterpart surface is also created (e.g., $S_{2,3,1}$).

External surface triangulation

The analytical model obtained until this point is already topologically consistent. Nevertheless, there is still one detail to be considered: some existing simulation tools (e.g., Energy Plus, tested herein) require convex polygons as input, at least for external surfaces. This feature is set as optional, depending on the needs of the target tool.

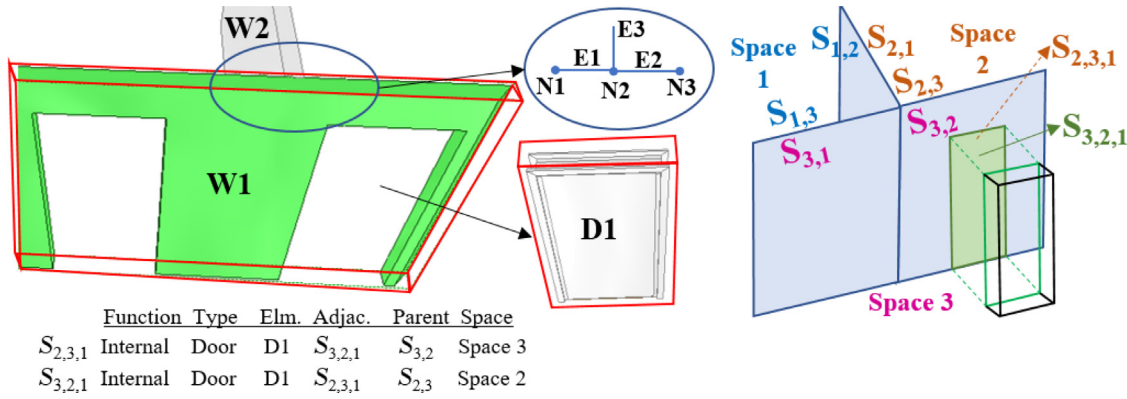


Fig. 19. Creation of opening surfaces.

The most appropriate way to do that is to split the concave polygon in triangles, because triangulation is a common issue in many fields and, consequently many free libraries exist [54,55]. Fig. 20 shows three different geometries from the same IFC model, focused on the roof: (a) the original solid geometries, (b) the analytical 2D surfaces (split according to the spaces found below) and (c) the previous surfaces after the triangulation of concave surfaces.

Creation of shadowing surfaces

All the surfaces generated in the previous steps correspond to surfaces which bound analytical spaces. Still, for an energy analysis perspective, it is highly relevant to identify the presence of other kind of surfaces which act as obstacles or shadowing surfaces and have an impact in the final energy balance, although they are not part of any space. The procedure developed covers a wide range of situations, which are displayed in Fig. 21.

First, let us consider balconies, overhangs, or any kind of salient of the façade. Two situations can occur:

- i. The element is part of an enclosure bounded with walls, curtain walls, etc. This means that wall instances exist forming a loop around the storey. Thus, the space under study is internal, as in (a), i.e., the inside area of the blue boundary. This situation has already been covered in previous steps.
- ii. The element is not bounded, i.e., it is in the open-air environment, as in (b). From the graph perspective its geometry is outside the loop of the storey, thus it is in fact external to the building, and the surfaces of its bounding box (6 in total) are converted to shadowing analytical surfaces ($S_{sh,j}$).

Additionally, already created surfaces belonging to unconnected ends (c) could still be present, as introduced in Fig. 14(c). In this case whether the edge (in red) is placed in or out of the storey loop

(in blue) must be checked. In this second case, a shadowing surface is also created.

Finally, the model could contain external buildings or urban elements, as in (d). Their geometry must be compared to the boundary loop of the storey they are assigned to, and, when necessary, the analytical shadowing surfaces $S_{sh,k}$ will be created.

Analytical model optimisation: Visibility analysis

The step presented now is not mandatory for the analytical model creation, but it has proven to be valuable for the case that the model spans over several buildings (e.g., a district) thus, when having a significant number of shadowing surfaces created as in the previous step. To make this algorithm efficient and extensible to the multiscale domain an optimisation technique has been implemented for reducing the number of shadowing surfaces to leave only the effective ones, i.e., the ones that are visible from the building under analysis.

Tests with Energy Plus engine have been made to justify this decision; in any case, it can be easily seen that:

- Non-visible surfaces cannot affect solar analysis (radiation masks or shading)
- Non-visible surfaces cannot affect energy analysis (shadows) nor acoustic analysis (noise barriers) since their impact is cancelled by another obstacle in-between.

Additionally, in earlier stages the geometry of all the objects in the IFC model has been parsed and their minimal bounding boxes calculated (as seen in Fig. 4). Let us assume the following (Fig. 22):

- Each bounding box is composed by 6 polygons, but only the four vertical sides are relevant.
- Each side has a normal vector pointing outwards the volume. If not, the polygon is reversed to guarantee that the requirement is met.

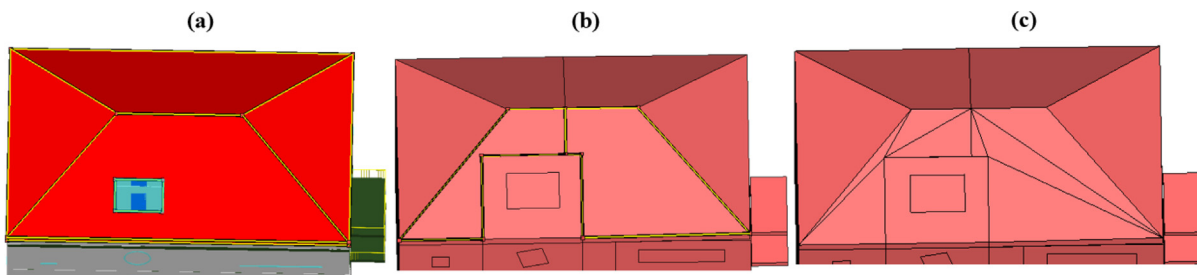


Fig. 20. Triangulation of external surfaces.

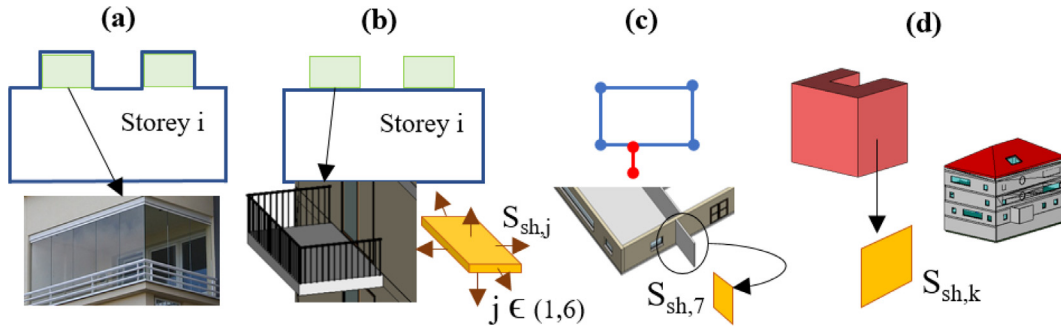


Fig. 21. Creation of shadowing surfaces.

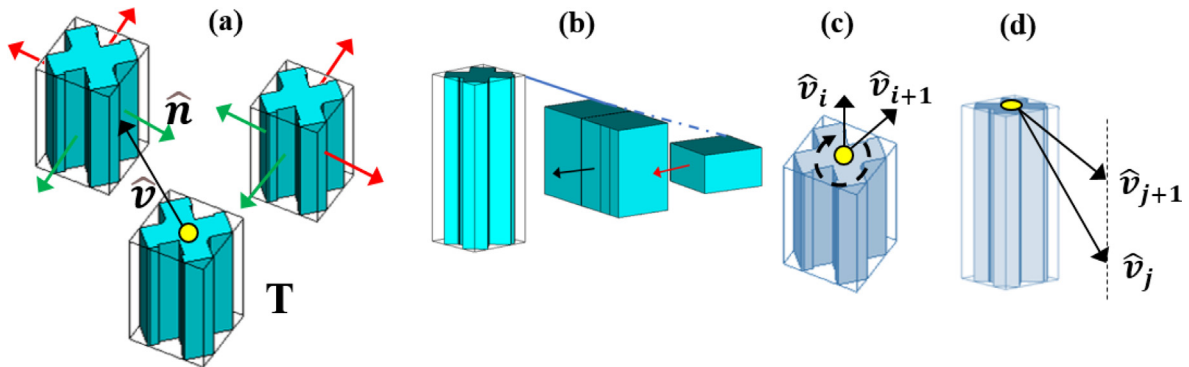


Fig. 22. Visibility analysis taken from a real pilot case in BIM4REN project [56].

- A side will only be visible from the target building if the normal point towards the building in the view direction (mathematically, if the dot product $\hat{n} \cdot \hat{v} < 0$, being \hat{n} the normal of the surface analysed and \hat{v} the vector from the target building to the surface centroid. All normal vectors that meet this condition in (a) are coloured in green, and the ones which not in red (consequently removed from the model), being the target building denoted with T.
- Even if the previous condition is met, a side (wall) is discarded if there is another polygon obstructing the trajectory, as in (b) where the wall with the red normal vector is obstructed by the central building.
- The maximum visibility from the target building is reached from the roof, thus, calculating the visible surfaces from each of the 4 top points of the bounding box and adding them all, assures that no surface goes undetected.
- This process must be repeated by iterating the vector \hat{v} through different azimuth angles (c) and elevations (d) and gathering all the surfaces detected in all the iterations (discarding repetitions).

To validate that this approach really enhances the simulation performance of a multi-building model, the following test has been conducted. A dummy Energy Plus base file has been created with a single building (Fig. 23). Then, several variations have been produced by changing the number (n) of shading surfaces, with the condition that, in all cases only the closest one is visible from the building; thus, the simulation results are equal. However, Energy Plus is not able to discard the non-visible surfaces and computes all of them when looking for potential shadowing effects, leading to a very inefficient performance (Table 2). The rapid increase in simulation time is noticeable when more than 50 external surfaces are present.

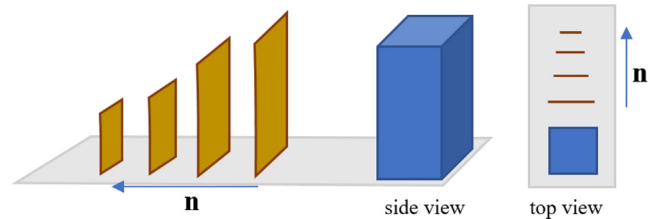


Fig. 23. Dummy input model for Energy Plus.

Fine-tuning of tolerances

Several of the steps followed in the methodology rely on some geometry tolerances and configuration values, being the result depending on these values. The application of threshold values to the previous two complete tests, but also many other tests done with other simpler or partial models, have served to adjust the values of the tolerances.

For instance, distance t in Fig. 5 has been set to 0.1 m. In general terms, a value close to the thickness of internal partitions have produced better results. Likewise, in the process of discarding extra or “noise” polygons generated during boolean operation processes (Fig. 18) the threshold values for polygon compactness have been set to $C_t = 0.1$ and $A_t = 1.0 \text{ m}^2$.

Some general conclusions extracted in relation to this adjustment process:

- Regular buildings (e.g., residential blocks) are likely to produce better results with no need to make adjustment using tolerances, whereas irregular buildings will produce a huge number of surfaces, which will probably need some cleaning or adjusting.

Table 2
Simulation times (in Energy Plus) for different number of shadowing surfaces.

Number of surfaces (n)	1	10	20	50	100	1000
Simulation time (seconds)	6	6	8	19	111	459

- BIM models from early design stage are also more likely to be more robust and require less algorithmic postprocessing, whereas as built or construction models are much more detailed and require filtering and cleaning non-relevant objects for thermal purposes, which is prone to require tolerance adjustments.
- Models created from reality capture devices are usually algorithmically created and not by user design (although the user can review and adjust it) are prone to have more topologically unconnected or clashing geometries. In those cases, the process of cleaning and fixing the model can be more intensive and could not always produce the desired results. Additionally, in multi-storey buildings the merging of partial scan results into a single building is not always perfect and tolerances appear.

3.5. Step 5: Output creation and export

Original IFC postprocessing and enrichment

The development of current work has been mainly motivated by giving the possibility to work on almost any kind of input IFC, regardless their origin and level of information, so that it can universally be applicable with no strict exchange requirements (even without spaces). In any case, the information and functional decomposition already modelled in the input IFC file can provide very valuable insight to complement the work done by the procedure described. In the other direction, the information generated by the algorithms can also enrich the original model, so that any third-party tool which imports the IFC can benefit from it and thus improve the interoperability.

In the context of current work, the term “functional space” (named as FS_n in Fig. 24(a)) will be used to denote the volume serving for certain functions or purposes in the building and typically bounded by walls and slabs. This is usually the meaning attributed to the *IfcSpace* entity in IFC. Note that in some cases a single spatial enclosure can be split in more than one functional space (like FS_3 and FS_4 , which are not separated by a wall, but a virtual boundary represented by a dashed line).

On the other hand, Fig. 24(b) shows the analytical spaces AS_n generated, presenting three main differences with the functional ones on (a):

- The boundary of analytical spaces extends up to the wall axis (centre line of the wall) because they have been generated by processing the graph obtained by edge intersections. Thus, they are slightly bigger than their functional counterparts. In other words, analytical spaces touch each other, whereas functional ones don't, because the space occupied by the wall is not accounted for.
- Analytical spaces are usually meant for physical simulations (thermal in the context of this work), thus, in general, virtual boundaries can be ignored and FS_3 and FS_4 would be mapped to a single analytical space AS_3 .
- For the same reason, external functional spaces (FS_7) representing a patio, terrace, hall or any kind of area in direct exposure with the external conditions is discarded (it is not meant to be heated).

Considering all this, the postprocessing of functional spaces has been implemented as follows:

- If the IFC model contains *IfcSpace* entities, for each one (FS_i) the matching analytical space(s) AS_j are identified. This can be done by selecting a random point (e.g., the centroid) of FS_i and check if it is inside the polygon defined by AS_j . In typical situations, we will have mostly 1 to 1 mapping, with few exceptions, as already shown (2 to 1 for FS_3 and FS_4) or unmapped cases (FS_7).
- If the model does not have *IfcSpace* entities, for each analytical space AS_i the associated walls are taken and by detection, intersection and trimming of the inner walls segments a polygon is created which represents the functional space footprint (steps a, b, c in Fig. 25). This polygon is extruded until the next storey. The only special case would be the case of tilted roofs, where the extruded volume is clipped with the lowermost roof polygons.

Once analytical spaces are mapped to functional ones (regardless of if they already existed in the model or have been automatically created afterwards), the area and volume of the corresponding functional space (namely A_{FS} and V_{FS}) are calculated and assigned to it along with the analytical area and volume (namely A_{AS} and V_{AS}). In general, $A_{FS} < A_{AS}$ and $V_{AS} < V_{FS}$. These

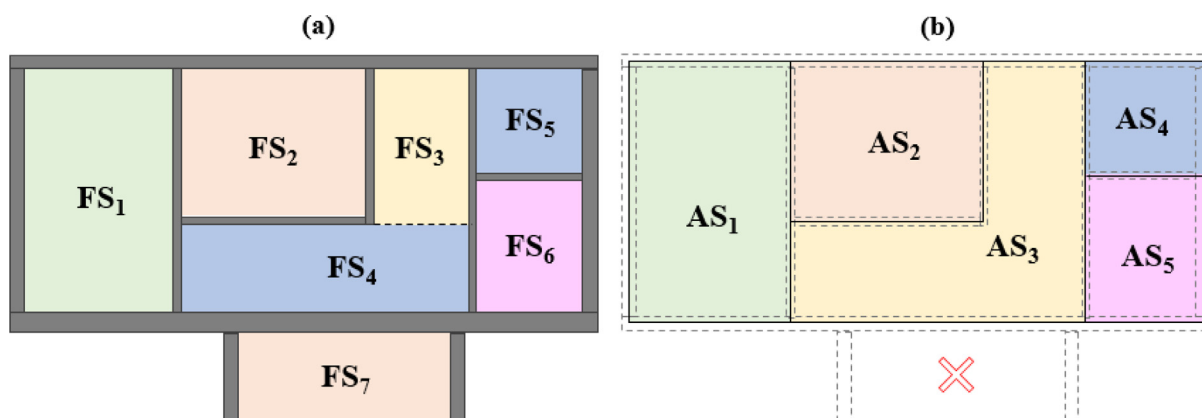


Fig. 24. Functional vs analytical spaces.

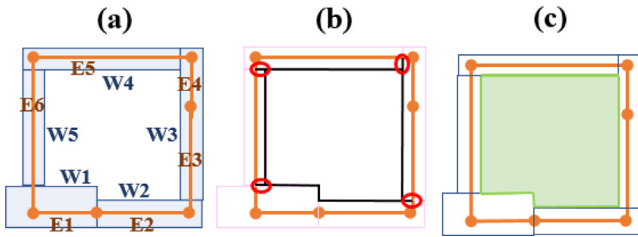


Fig. 25. Creation of functional spaces.

deviations could be considered for adjusting or normalizing the simulation results generated using the analytical spaces.

Creation (or overriding) of space boundaries and walls' connectivity

From the building topology perspective, there are two semantic concepts that are commonly used by many BIM software tools: space boundaries and walls' connectivity with other walls along a path. Both can be derived from the calculated analytical surfaces and from the node and edge configuration of the graphs.

In case the original IFC model already contains these features, the algorithm can be configured to leave the original ones or to override them by the calculated ones. The IFC enhancement works as depicted in Fig. 26, which is based on the examples presented in Fig. 19 and Fig. 6.

- Space boundaries, which are materialized through the `IfcRelSpaceBoundary2ndLevel` class (only after version IFC 4). Each analytical surface is converted to an `IfcRelSpaceBoundary2ndLevel` instance properly linking related building elements, spaces and adjacent surfaces (Fig. 26, a).
- Walls' connections to other walls are modelled through an `IfcRelConnectsPathElements` instance (Fig. 26, b), by mapping walls' start, end and middle points.

The original IFC file is enriched with all these relationships, being topologically more complete.

Export to IDF and gbXML

The goal of the described procedure is to create a ready-for-simulation file. Two approaches have been considered: on the one hand the export to gbXML, a BEM standard widely supported by the industry, thus ideal for data exchange with third-party energy simulation tools. And on the other hand, the export to IDF (the input file of Energy Plus, an open-source engine). The first

approach is more universal, the second one more specific, but it allows to automate the simulation process with Energy Plus, i.e., not only to create the input file but to invoke in the background one or more simulation runs by command-line.

Fig. 27 shows the mapping convention followed in each case. In the middle, the previously explained analytical model is depicted (consisting of analytical zones, spaces and surfaces), with their adjacency and parent relationships. On the left, the basic gbXML entities are shown (focused only on the geometry and topology, i.e., systems are not yet covered). Dashed lines represent the concept mapping. An equivalent strategy has been followed with the IDF model (on the bottom). Then, on the right side a sample IDF file fragment is shown, which illustrates these concepts.

It may be noticed that, even if all the models follow a similar approach, the implementation is slightly different and mapping is not always straightforward. For instance, in the internal analytical model there is a single "surface" concept (which can be of type "wall", "floor", "ceiling", "door", "window", "roof" or "shadow"). In gbXML opaque (e.g., wall) and opening (e.g., window) surfaces are modelled using two different entities; also, in IDF (building vs fenestration) where we have an additional shading surface not existing in gbXML (where a shading is just a type of opaque surface). Similarly, in gbXML zones and spaces are two different concepts, whereas in IDF only the concept of zone exists (although they can be grouped). Their hierarchy itself (denoted by black arrows) is slightly different.

All the test cases conducted to validate this approach have been limited to the geometry/topology and single space zones (i.e., each space is an individual analytical zone).

4. Testing of the methodology

4.1. Application to test buildings

To validate the developed methodology-two complementary test cases have been conducted.

The first test case consists of an ad-hoc created complete building model, with its surroundings, using Autodesk Revit design tool. It has been created in such a way that presents almost all kinds of special situations described in this work (as shown in Fig. 28), namely: (1) a collection of external buildings modelled as generic blocks, some of them occluded from the main building. The target building consists of 4 storeys above ground and one below, with no spaces modelled inside and with different wall partitions layout in each storey. Additionally, there are balconies (2), unconnected

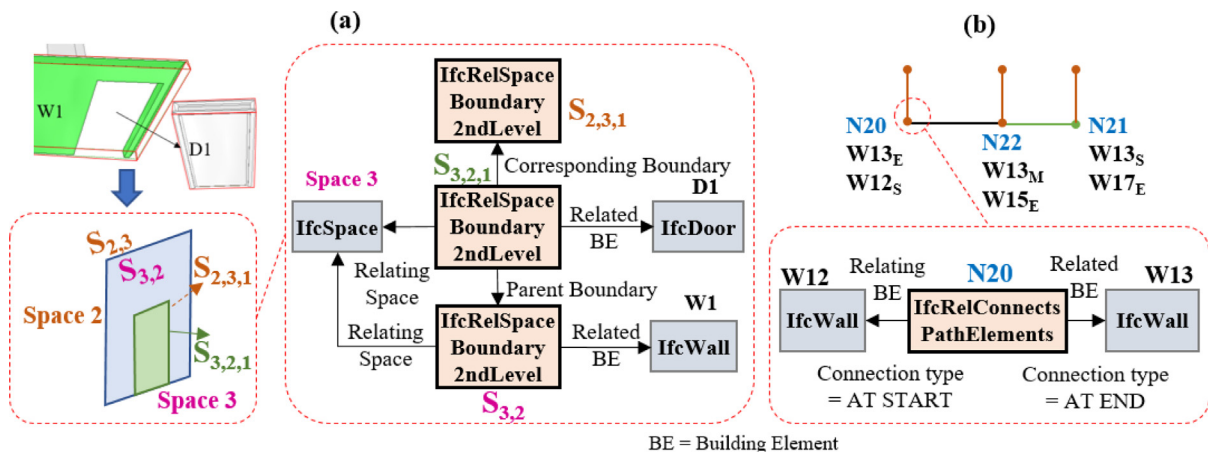


Fig. 26. Creation of space boundaries and walls' connectivity.

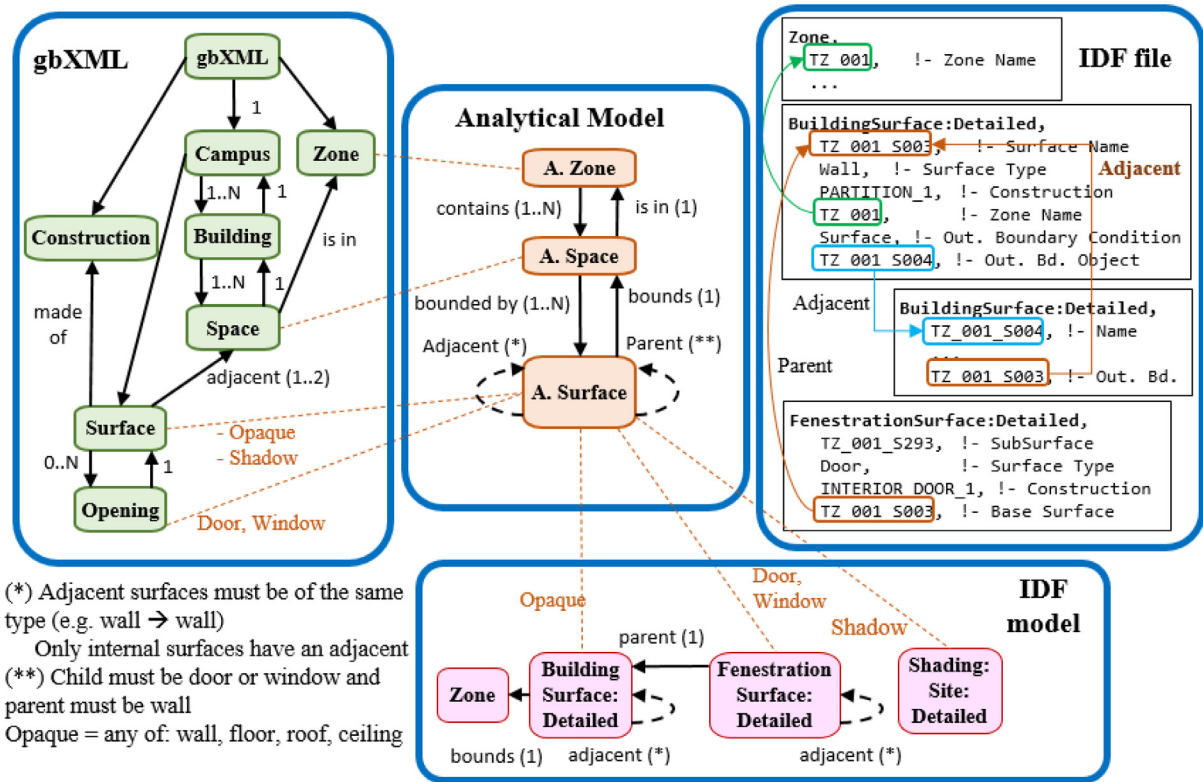


Fig. 27. Export to different formats.

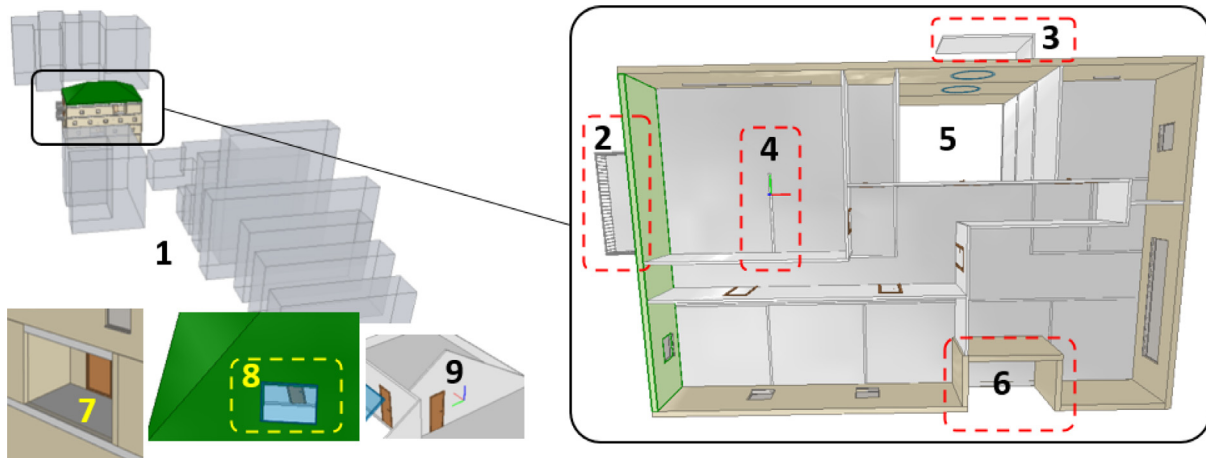


Fig. 28. Test case 1: ad-hoc created model.

walls both externally (3) and internally (4), a void which spans all the storeys (5), façades with setbacks (6 or 7), windows in the roof (8) not associated to a wall, and not modelled with a IfcRelFills relationship, and, finally, tilted roof surfaces yielding not rectangular walls in the last storey (9).

With this model a complete generation of graphs, spaces and surfaces has been conducted, yielding the results shown in Table 3. Since the input model does not contain spaces, these have been auto generated from scratch. To illustrate the comprehensiveness of the loop search process, it is noticeable that even with a small number of nodes and edges as in Storey 1, the number of total loops created by brute force is higher than 10 000. Then the subsequent optimisation process reduces drastically this amount, ending up with 10 loops to be converted to spaces. The whole process took

less than a minute, which is a very competitive value considering the time that a designer must spend to properly model spaces and exporting or creating an energy model from scratch.

Although strictly speaking this is not a necessary step, this internal analytical model was also converted to an “Analytical IFC” (Fig. 29). The reason is that in the case of surface types and function it contains a richer variety than the one in gbXML, IDF or even in the IFC space boundaries. This provides a visual and easily understandable log of the algorithm output, which aids in verifying the correctness of the outputs. This IFC only contains IfcSpace entities and thermal surfaces created as IfcBuildingElementProxy entities with several properties for type, thermal function, adjacent object, etc. and 2D surface geometry. Several free IFC viewers support the model auditing and colouring by property value, e.g., FZK

Table 3
Results of the graph generation process per storey.

Storey	-1	0	1	2	Roof	Total
Number of walls	7	16	17	18	8	66
Number of nodes (all)	8	22	27	25	10	92
Number of nodes (valid, after merge)	8	20	27	23	10	88
Number of edges	9	25	36	29	12	111
Number of loops (all, brute force)	31	729	10,943	1565	80	13,348
Number of loops (valid polygons)	8	230	4104	482	24	4848
Number of loops (unique, repetitions removed)	3	47	537	85	7	679
Number of loops (minimal, the real spaces)	2	6	10	7	3	28
Number of analytical surfaces (original)						298
Number of analytical surfaces (valid)						283

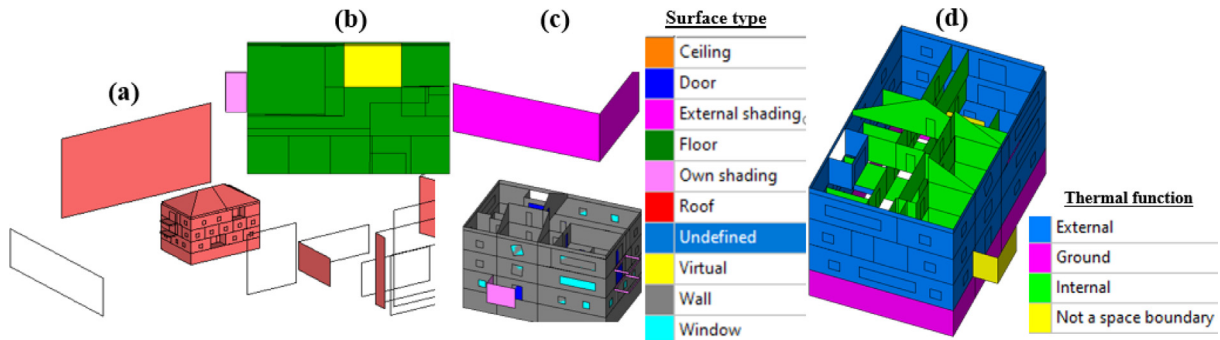


Fig. 29. Internally created analytical IFC file.

Viewer [57], used in this example. This is much more helpful than a textual log file and errors (e.g., external surface marked as internal) can very easily be detected. Also, the number of shadowing surfaces created has been reduced to strictly the ones affecting the target building (see Fig. 29 (a), where transparent polygons mean that we are seeing them from the opposite side of their normal vector). Surfaces acting as thermal boundary of a space are properly distinguished of those acting as own or external shadows or virtual space separators.

The second complete test case is on the one hand simpler in terms of variety (no external buildings, no balconies and recesses, no virtual boundaries, etc.), but, on the other hand, it was produced by a commercial Scan-2-BIM application ARtoBuild by Bimeo [46], which generates an IFC model from an on-site scanning done using an iPad, as part of the H2020 EPC-RECAST project [58]. This means that since the IFC model is mostly generated by an algorithm and not by manual design, is prone to have situations more difficult to handle in the graph detection, such as overlapping walls or

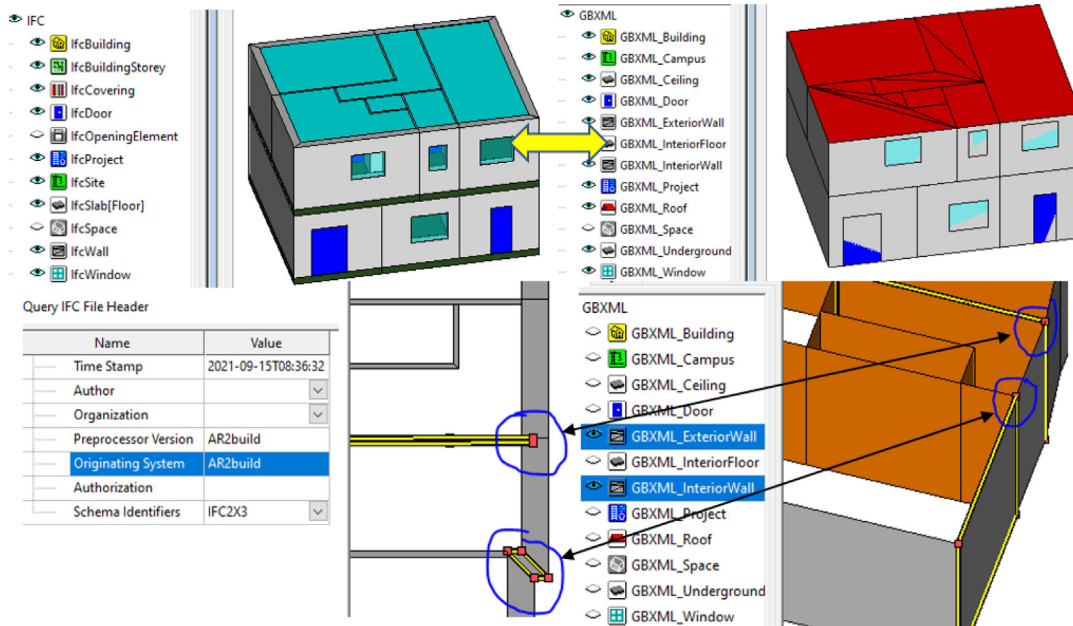


Fig. 30. Test case 2: model generated through a mobile scanning process.

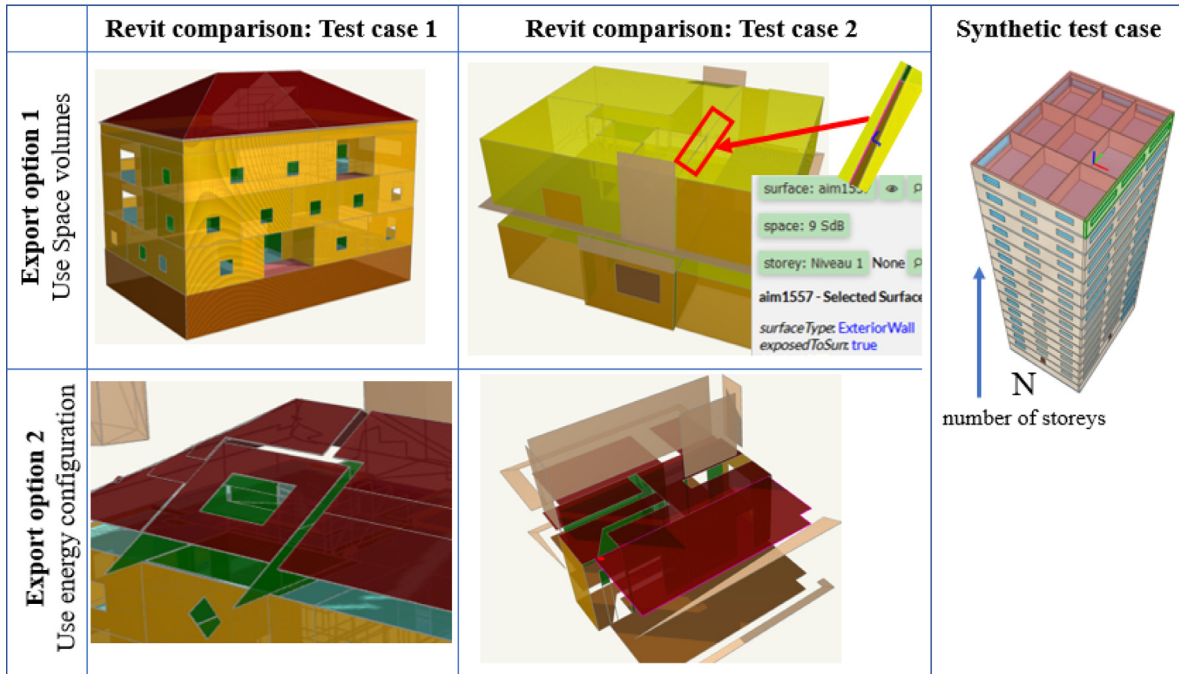


Fig. 31. Results of the Revit to gbXML export of the two test cases.

redundant wall segments created (as shown in Fig. 30, bottom, denoted by circles). As it is shown, the developed algorithm has been able to successfully fix this and create cleanly connected surfaces. Fig. 30 also shows in the right side, the complete building modelled converted to gbXML.

Although still more comprehensive tests will be performed, these initial results are very promising, since the algorithm has produced successful outputs using IFC models from different origins and different quality. This is especially interesting when it comes to applying the algorithm to renovation processes (e.g., when auditing the building for Energy Performance Certificate issuing), where usually no design BIM models exist and, in general, the input BIM source will probably consist of the output of a Scan-2-BIM process. This can undoubtedly help in automating the process.

4.2. Comparison with existing methods

To illustrate the effectiveness of the described method a comparison with existing methods has been done. For that, Autodesk Revit has been selected, since it is one of the most widely used BIM authoring tool, which includes built-in gbXML export capabilities, with two export options: (1) Use space volumes and (2) Use energy configuration. Fig. 31 (left and middle columns) show the results of gbXML exports of both test cases presented before, using each one of the export options (4 exports in total). The displayed screenshots have been taken from Spider gbXML Viewer [59], an online tool for viewing and navigation into gbXML files.

No significant differences were found in processing time (Revit took less than 10 s in both cases). However, the quality of the generated files with Revit was quite unsatisfactory, using any of the two export configuration options available:

- Option 1 creates a better result. However, it requires a very careful space modelling by an expert, and even in this case errors can appear, as Test Case 2, where several gaps and overlaps appear, which provoke some internal walls to be identified as external. It does not export either any of the external build-

ings, overhangs, balconies or similar elements, whereas the algorithm proposed in current work was able to detect those elements, e.g., Fig. 29, (c) and (d).

- Option 2 performs a deeper detection of surfaces, including externals, without considering space enclosures, but the quality is notably worse, since they are not properly joined and closed.

To sum up, none of these results are satisfactory and can be considered for a valid comparison with the method proposed. Additionally, they require introducing specific commercial tools in the workflows.

Additionally, a performance test has been conducted using a batch of synthetically generated IFC files (Fig. 31, right), which serves to benchmark the algorithms with respect to varying input model sizes. The layout is quite simple, a storey divided in 9 internal spaces. In each iteration of the batch run, an additional storey was added to the IFC. Fig. 32 shows the increase of computation time by varying the number of storeys from 1 to 25, considering the whole process of generating the analytical model and exporting to gbXML. Even in the most extreme case (25 storeys, more than 200 spaces) it only took 75 s. It is in any case a geometrically very simple and regular model.

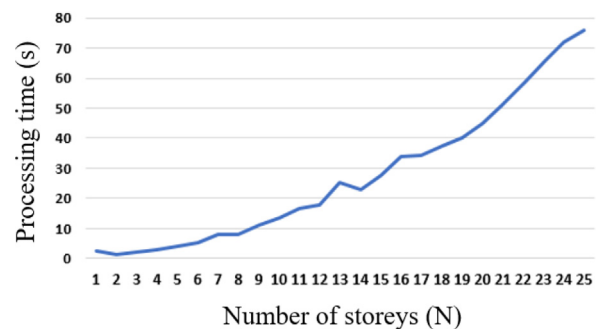


Fig. 32. Performance tests with the synthetic IFCs.

5. Conclusions and next steps

The procedure presented in this paper exploits the potential of graphs and computational geometry processing with the aim to drastically reduce the requirements of the input IFC model for analytical model generation of a multi-building complex, at least from the geometrical perspective.

Thus, the results obtained in the testing phase have been successful in relation to the main objectives pursued:

- In contrast to similar approaches, the methodology can handle not proper geometries (not properly connected or overlapping) not even requiring physical spaces to be modelled in IFC, thus the end-user expertise is reduced to minimum. This is of utmost relevance when IFC models are not created by user modelling, but algorithmically by means of reality capture techniques (3D scanning or AI-based object recognition in drawings), being more likely to have problematic geometries.
- It also considers the impact of the building surroundings and efficiently simplifies its possible complexity.
- It also goes beyond traditional BIM modelling tools when creating the space boundaries, by enriching them with additional information (surface type, thermal function, distinction of shadowing vs thermal surfaces, removal of not relevant surfaces, etc.).

The obtained results are already promising with very competitive computational values achieving just a few seconds. This approach can help to automate building design and renovation workflows, by seamlessly integrating energy simulation capabilities for non-experts, reducing time and costs in the decision-making process, and universalize the use of energy simulation within any open BIM workflow.

However, in the performed validation, some potential limitations have been identified in relation to the geometry processing, which are still being enhanced:

- The algorithms rely on multiple mathematical operations (node alignment, boolean operations of polygons, identification of adjacencies and intersections, etc.), which, as explained in the paper, depend in many tolerances and internal configuration values. For BIM models coming from end-user designs, the algorithms are currently quite robust, but for models derived from reality capture process, they are still sensitive to the choice of tolerance values and configuration parameters chosen.
- Tests models used represent residential blocks, whose geometry variety is limited. Very irregular and complex geometries still require careful tests implying a higher complexity of the graph size, and the tolerance values selection could have greater impact.

These aspects are consequently subject of future research directions, such as:

- Consider IFC inputs coming from different reality capture tools, in addition to ARtoBuild, e.g., IFC models generated from vision-based AI processing of floorplans from a PDF or picture, such as plans2BIM [60]. Thus, it must be verified that the assumptions done, and tolerances selected can also be universally applicable to any IFC generation tool.
- Implementation of more advanced error postprocessing capabilities (as done by Ying and Lee [61,62]) seems a natural evolution. For example, provide auto-adjust capabilities (i.e., to perform different runs in loop by varying the tolerance values until the best configuration is found, i.e., the one with no

errors). This could, on the other hand, increase the computation time, so it could be introduced as an optional and configurable feature.

- Also, for contextual information gathering (especially the geometry of surroundings) there is an increasing number of publicly available data sources (local cadastres, OpenStreetMap, etc.) which can be used as input for the surroundings modelling. Those datasets are currently being considered in ongoing research lines.

Finally, although the core of the work is related to the geometry and topology, which are the most complex aspects to automate, it is on the scope of the authors to extend to algorithms to other aspects:

- In relation to the elements' composition and semantics, for each category of the thermal surfaces (external, ground, internal, etc.) default thermal properties have been chosen when converting to IDF or gbXML materials and construction techniques. In future versions, this can be directly read from the properties in IFC (when existing), integrating a product library or even by connecting to cloud-based libraries or product data catalogues, e.g., linking to bsDD (buildingSMART Data Dictionaries) [63] which already provides an API.
- In relation to energy systems (heating, cooling and ventilation or lighting and electrical devices), although BEM data models (e.g., gbXML or IDF analysed herein) support them, the research lines on the automated BIM-2-BEM interoperability on these topics is less mature. Additionally, if we focus on renovation processes and reality capture workflows, they are mostly focused on the geometry generation and the networks topology and connectivity is rarely addressed or poorly defined in IFC which makes it more difficult. However, since technology is rapidly evolving, monitoring of new innovations in the market should be done, so that they can be integrated in automated workflows.
- In the meantime, some of the temporary limitations can be solved by implementing easy-to-use user interfaces for enriching the IFC model with missing information.

Data availability

No data was used for the research described in the article.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Asier Mediavilla reports financial support was provided by European Commission.

The remaining 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.

Acknowledgements

The authors would like to express the gratitude to the European Commission by funding the research projects BIM4REN, EPCRE-CAST and ENSNARE (Grant Agreement No. 820773, 893118 and 958445, respectively), under the Horizon 2020 programme, where the presented work was conducted. This manuscript reflects only the authors' views, and the Commission is not responsible for any use that may be made of the information it contains.

References

- [1] European Commission (2022). Energy, Climate change, Environment, Overall targets and reporting. https://ec.europa.eu/info/energy-climate-change-environment/overall-targets-and-reporting_en.
- [2] European Commission (2002). Directive 2002/91/EC on the Energy Performance of Buildings. <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2003:001:0065:0071:EN:PDF>.
- [3] European Commission (2021). Proposal for a Directive on the energy performance of buildings (recast) COM/2021/802 final. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0802&qid=1641802763889>.
- [4] L.F. Cabeza, M. Chàfer, Technological options and strategies towards zero energy buildings contributing to climate change mitigation: a systematic review, *Energy Build.* 219 (2020), <https://doi.org/10.1016/j.enbuild.2020.110009>.
- [5] M.L. Rivera, H.L. MacLean, B. McCabe, Implications of passive energy efficiency measures on life cycle greenhouse gas emissions of high-rise residential building envelopes, *Energy Build.* 249 (2021), <https://doi.org/10.1016/j.enbuild.2021.111202>.
- [6] M.A. Mujeebu, F. Bano, Energy-saving potential and cost-effectiveness of active energy-efficiency measures for residential building in warm-humid climate, *Energy Sustain. Dev.* 67 (2022) 163–176, <https://doi.org/10.1016/j.esd.2022.01.011>.
- [7] A. Ahmed, T. Ge, J. Peng, W.C. Yan, B.T. Tee, S. You, Assessment of the renewable energy generation towards net-zero energy buildings: a review, *Energy Build.* 256 (2022), <https://doi.org/10.1016/j.enbuild.2021.111755>.
- [8] Asdrubali, F., Desideri, U. (Ed.) (2018) Handbook of Energy Efficiency in Buildings, A Life Cycle Approach, 1st Edition, Elsevier, ISBN 9780128128176, <https://doi.org/10.1016/B978-0-12-812817-6.09989-7>.
- [9] European Construction Sector Observatory. (2019). Building Information Modelling in the EU Construction Sector. Trend Paper Series. <https://segm.gr/wp-content/uploads/2019/03/ECSO-TP-BIM-in-construction-March-2019.pdf>.
- [10] World Economic Forum. Shaping the Future of Construction. 2016, p. 5. Available online: http://www3.weforum.org/docs/WEF_Shaping_the_Future_of_Construction_full_report_.pdf.
- [11] Official Journal of the European Union, 2014–EUR-Lex. Available online: <https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=celex%3A32014L0024>.
- [12] European Construction Sector Observatory. (2021) Digitalisation in the Construction Sector. Analytical Report. <https://ec.europa.eu/docsroom/documents/45547>.
- [13] Z. Liu, Y. Lu, L.C. Peh, A review and scientometric analysis of global building information modeling (BIM) research in the architecture, engineering and construction (AEC) industry, *Buildings* 9 (2019) 210, <https://doi.org/10.3390/buildings9100210>.
- [14] M. Sagarna, J.P. Otaduy, F. Mora, I. Leon, Analysis of the state of building conservation through study of damage and its evolution with the state of conservation assessment BIM model (SCABIM), *Appl. Sci.* 12 (2022) 7259, <https://doi.org/10.3390/app12147259>.
- [15] E. Kamel, A.M. Memary, Review of BIM's application in energy simulation: tools, issues, and solutions, *Autom. Constr.* 97 (2018) 164–180, <https://doi.org/10.1016/j.autcon.2018.11.008>.
- [16] M. Elagiry, N. Charbel, P. Bourreau, E. Di Angelis, A. Costa. (2020). IFC to building energy performance simulation: a systematic review of the main adopted tools and approaches. *BauSIM 2020 - 8th Conference of IBPSA Germany and Austria*. <https://doi.org/10.3217/978-3-85125-786-1-63>.
- [17] buildingSMART International. (2020). *Information Delivery Manual*. <https://technical.buildingsmart.org/standards/information-delivery-manual> (Accessed 15 July 2022).
- [18] buildingSMART International. (2020). *Model View Definitions*. <https://technical.buildingsmart.org/standards/ifc/mvd/> (Accessed 15 July 2022).
- [19] S. Pinheiro, R. Wimmer, J. O'Donnell, S. Muhic, V. Bazjanac, T. Maile, J. Frisch, C. van Treeck, MVD based information exchange between BIM and building energy performance simulation, *Autom. Constr.* 90 (2018) 91–103, <https://doi.org/10.1016/j.autcon.2018.02.009>.
- [20] buildingSMART International. (2020). *Industry Foundation Classes Specification, version 4.0*. Available at https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/ (Accessed 15 July 2022).
- [21] buildingSMART International. (2022). *Information Delivery Manual (IDM) Development for Building Information Modelling (BIM) and Building Energy Modelling (BEM) Workflows*. <https://www.buildingsmart.org/bim-bem-workflows-final-technical-report-approved-by-sc-and-published/> (Accessed 15 December 2022).
- [22] H. Gao, C. Koch, Y. Wu, Building information modelling based building energy modelling: a review, *Appl. Energy* 238 (2019) 320–343, <https://doi.org/10.1016/j.apenergy.2019.01.032>.
- [23] R. Alsharif. (2019). A review on the challenges of BIM-based BEM automated application in AEC industry. <https://doi.org/10.13140/RG.2.2.26102.55366>.
- [24] H. Fernald, S. Hong, L. Liam O'Brien, S. Bucking. (2018). BIM to BEM translation workflows and their challenges: a case study using a detailed BIM model. *Proceedings of eSim 2018, the 10th conference of IBPSA-Canada*, pp. 482–491.
- [25] I.J. Ramaji, R. Williams, J.I. Messner, E. Mostavi, IFC-based BIM-to-BEM model transformation, *J. Comput. Civ. Eng.* 34 (3) (2020), [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000880](https://doi.org/10.1061/(asce)cp.1943-5487.0000880).
- [26] A. Armijo, P. Elguezabal, N. Lasarte, M.A. Weise, Methodology for the digitalization of the residential building renovation process through OpenBIM-Based workflows, *Appl. Sci.* 11 (21) (2021) 10429, <https://doi.org/10.3390/app112110429>.
- [27] Autodesk (2022). Autodesk Revit. Available at <https://www.autodesk.com/products/revit/overview> (Accessed 15 July 2022).
- [28] Trimble (2022). SketchUp. Available at <https://www.sketchup.com/> (Accessed 14 November 2022).
- [29] U.S. Department of Energy. (2022). *Energy Plus*. Available at <https://energyplus.net/> (Accessed 15 July 2022).
- [30] Green Building XML (2022). *gbXML schema*. Available at: <https://gbxml.org> (Accessed 15 July 2022).
- [31] C.M. Rose, V. Bazjanac, An algorithm to generate space boundaries for building energy simulation, *Eng. Comput.* 31 (2) (2015) 271–280, <https://doi.org/10.1007/s00366-013-0347-5>.
- [32] 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 (2022) 221, <https://doi.org/10.3390/buildings12020221>.
- [33] A. Mediavilla, X. Arenaza, V. Sánchez, Y. Sebesi, P. Philipps, in: *RenoBIM: Collaboration Platform Based on Open BIM Workflows for Energy Renovation of Buildings Using Timber Prefabricated Products*, Taylor & Francis Group, 2018, pp. 281–288, <https://doi.org/10.1201/9780429506215-35>.
- [34] G.N. Liliis, G.I. Giannakis, D.V. Rovas, Automatic generation of second-level space boundary topology from IFC geometry inputs, *Autom. Constr.* 76 (2017) 108–124, <https://doi.org/10.1016/j.autcon.2016.08.044>.
- [35] T. El-Diraby, T. Krijnen, M. Papagelis, BIM-based collaborative design and socio-technical analytics of green buildings, *Autom. Constr.* 82 (2017) 59–74, <https://doi.org/10.1016/j.autcon.2017.06.004>.
- [36] D. Ladenhauf, K. Battisti, R. Berndt, E. Eggeling, D.W. Fellner, M. Gratzl-Michlmair, et al., Computational geometry in the context of building information modelling, *Energy Build.* 115 (2016) 78–84, <https://doi.org/10.1016/j.enbuild.2015.02.056>.
- [37] W. Jeong, J. Son, An algorithm to translate building topology in building information modeling into object-oriented physical modeling-based building energy modeling, *Energies* 9 (1) (2016) 50, <https://doi.org/10.3390/en9010050>.
- [38] H. Ying, S. Lee, Generating second-level space boundaries from large-scale IFC-compliant building information models using multiple geometry representations, *Autom. Constr.* 126 (2021) 103659.
- [39] C. Van Treeck, E. Rank, Dimensional reduction of 3D building models using graph theory and its application in building energy simulation, *Eng. Comput.* 23 (2007) 109–122, <https://doi.org/10.1007/s00366-006-0053-7>.
- [40] H. Chen, Z. Li, X. Wang, B. Lin, A graph- and feature-based building space recognition algorithm for performance simulation in the early design stage, *Build. Simul.* 11 (2) (2018) 281–292, <https://doi.org/10.1007/s12273-017-0412-x>.
- [41] Solibri. (2022). *Solibri tools*. Available at: <https://www.solibri.com/> (Accessed 15 July 2022).
- [42] R.J. Trudeau, *Introduction to Graph Theory*, Dover Pub, New York, 1993.
- [43] K. Urrutia-Azcona, M. Tatar, P. Molina-Costa, I. Flores-Abascal, *Cities4ZERO: overcoming carbon lock-in in municipalities through smart urban transformation processes*, *Sustainability* 12 (9) (2020) 3590.
- [44] K. Urrutia-Azcona, E. Usobiaga-Ferrer, P. De Agustín-Camacho, P. Molina-Costa, M. Benedito-Bordonau, I. Flores-Abascal, ENER-BI: integrating energy and spatial data for cities' decarbonisation planning, *Sustainability* 13 (2021) 383, <https://doi.org/10.3390/su13010383>.
- [45] IEA. International Energy Agency (2013). *Transition to Sustainable Buildings. Strategies and Opportunities to 2050*. https://iea.blob.core.windows.net/assets/1e300ab6-44de-41dc-8714-ee12a4800943/Building2013_free.pdf.
- [46] Bimeo (2022). *ARtoBuild*. Available at: <https://www.bimeo.fr/artobuild> (Accessed 15 July 2022).
- [47] R. Graham, An efficient algorithm for determining the convex hull of a finite planar set, *Inf. Process. Lett.* 1 (1972) 132–133, [https://doi.org/10.1016/0020-0190\(72\)90045-2](https://doi.org/10.1016/0020-0190(72)90045-2).
- [48] G. Toussaint. Solving geometric problems with the rotating callipers. *Proceedings of the IEEE*, Athens, Greece, 1983.
- [49] D. Eberly. *Minimum-Area Rectangle Containing a Set of Points*. Available at: <https://www.geometrictools.com/Documentation/MinimumAreaRectangle.pdf> (Accessed 15 July 2022).
- [50] buildingSMART International. (2020). *Second-level spaces boundaries in IFC*. Available at https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/ink/ifcrespaceboundary2ndlevel.htm (Accessed 15 July 2022).
- [51] D. Menegatti. (2021). *Polybool-Java: open-source Java library for boolean operations on polygons*. Available at: <https://github.com/Menegatti/polybool-java> (Accessed 15 July 2022).
- [52] F. Martínez, A.J. Rueda, F.R. Feito, A new algorithm for computing boolean operations on polygons, *Comput. Geosci.* 35 (6) (2009) 1177–1185, <https://doi.org/10.1016/j.cageo.2008.08.009>.
- [53] W. Li, M.F. Goodchild, R. Church, An efficient measure of compactness for two-dimensional shapes and its application in regionalization problems, *Int. J. Geogr. Inf. Sci.* 27 (6) (2013) 1227–1250, <https://doi.org/10.1080/13658816.2012.752093>.
- [54] J. Simomaa. *Earcut4j: open-source Java library for polygon triangulation*. Available at: <https://github.com/earcut4j/earcut4j> (Accessed 15 July 2022).

- [55] D. Eberly. *Triangulation by Ear Clipping*. Available at: <https://www.geometrictools.com/Documentation/TriangulationByEarClipping.pdf> (Accessed 15 July 2022). 2002.
- [56] BIM4REN. Horizon 2020 Project, Grant Agreement No. 820773; European Commission: Brussels, Belgium. Available online: <https://bim4ren.eu> (accessed on 11 October 2022).
- [57] Karlsruher Institut für Technologie. (2022). *FZK Viewer*. Available at <https://www.iai.kit.edu> (Accessed 15 July 2022).
- [58] EPC-RECAST Project. Horizon 2020 Project, Grant Agreement No. 893118; European Commission: Brussels, Belgium. Available online: <https://epc-recast.eu> (accessed on 11 October 2022).
- [59] Spider gbXML Viewer (2022). Available at: <https://www.ladybug.tools/spider-gbxml-tools/spider-gbxml-viewer/v-0-17-08/app-viewer/spider-gbxml-viewer.html>. (Accessed 8 November 2022).
- [60] WiseBIM (2022). *Plans2BIM*. Available at: <https://plans2bim.com/> (Accessed 15 July 2022).
- [61] H. Ying, S. Lee, Automatic detection of geometric errors in space boundaries of IFC-BIM models using monte carlo ray tracing approach, *J. Comput. Civ. Eng.* 34 (2) (2020), [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000878](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000878).
- [62] 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>.
- [63] buildingSMART International. (2022). *buildingSMART Data Dictionaries (bsDD)*. Available at <https://www.buildingsmart.org/users/services/buildingsmart-data-dictionary/> (Accessed 15 July 2022).