EXTRACTING SPATIAL STRUCTURE FROM BUILDING INFORMATION MODELS

STYLIANOS DRITSAS[1], KHYSTELLE YEO[2] and CHERYL LEE[3]
[1,2,3]Singapore University of Technology and Design.
[1]dritsas@sutd.edu.sg, 0000-0002-9609-2784
[2]khystelle_yeo@sutd.edu.sg 0009-0007-3478-0876
[3]cheryl_lee@sutd.edu.sg 0009-0001-0965-9368

Abstract. The objective of this work is to automatically extract the spatial structure from Building Information Models for subsequently performing building circulatory analysis and regulatory building code compliance checking. This article presents the model comprehension methods employed, the challenges faced, and the solutions developed. We highlight the conceptual and technical limitations of current BIM paradigms and discuss how they may be improved.

Keywords. Building Information Modelling, Automated Extraction, Spatial Structure.

1. Introduction

Building Information Modelling (BIM) gained acceptance during the past few decades within the Architecture, Engineering, and Construction (AEC) industry (Eastman et al, 2011). It offers a natively 3D and semantically rich paradigm of design representation compared with Computer Aided Design (CAD), namely 2D drawings (Howard and Andersen, 2001). Object-based modelling overcomes the limitations associated with conflating geometry, topology, and semantics found in drawings such as plans and sections. Nevertheless, there are still numerous obstacles associated with recovering the spatial structure of a design represented using BIM.

Spatial structure is the logical graph of relationships between a space and its components and among spaces. The difficulty of deriving spatial structure springs from a fundamental problem: space itself is never explicitly modelled but merely implied as the result of its bounding surfaces. Thus, the definition of space geometrically and topologically, in the sense of the relationships among its semantic components, is limited and challenging to reconstruct. If this data was available then building models would allow queries combining geometry, topology, and semantics, such as which elements are associated with a space, which spaces share the common elements, which space contains an element, and which spaces are adjacent or nested within one another.

In theory, and to a certain extent in practice, spaces are captured by such notions as room elements within current BIM, however, these concepts are often poorly implemented and severely limited. Indicatively, the concept of a building's urban context in the sense of its embedding within space is unrepresentable. Additionally,

ACCELERATED DESIGN, Proceedings of the 29th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2024, Volume 1, 231-240. © 2024 and published by the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Hong Kong.

determining the extents of spatial volumes that span several levels is problematic, expressing the structure of regions with an open plan space without physical boundaries is unclear, and buildings that conform to complex terrain conditions do not adhere to the conventional notion of a building level.

What would be beneficial for spatial analysis is a graph-theoretic object model where relationships are first-class concepts instead of the result of implicit queries among database tables and computationally expensive solid Boolean operations. This is a conceptual limitation of the BIM object model and its associated data structures (Eastman et al. 2010). Thus, today spaces are manually drawn in the plan, and this opens the door for challenges such as human errors which paired with a lack of industry modelling standards and quality control protocols result in inconsistencies that limit the utility of BIM (Migilinskasa et al. 2013; Wang et al. 2014; Chen and Luo, 2014).

2. Materials and Methods

The objective of the system developed was to automatically recover spaces and their relationships by examining a building information model. The application domain for the data is building circulation (Lee et al. 2010; Firas et al. 2023), feasibility analysis (Sherif and Eastman, 2011), and code compliance (Eastman et al. 2009; Balaban et al. 2012). The data required is semantically labelled regions and graphs conveying the relationship between those. Three basic region types are defined: (a) Spaces, capturing the notion of accessible, in the net internal floor plan area sense, within and around a building; (b) Accesses, expressing thresholds between spaces such as doors, elevators, escalators, and stairs. In addition, the category of virtual access is used for expressing portals where information is either unavailable or irrelevant, such as crossings to adjacent parcels; and (c) Visuals, contain information about BIM elements such as floors, walls, windows, columns, used for visualization (Fig. 1). Regions are expressed as surfaces bounded by piece-wise linear curves containing one exterior ring and optionally interior holes. Attributes associated with a region include a globally unique identifier, a reference to BIM elements, building level number and elevation, and categorical information such as usage information.

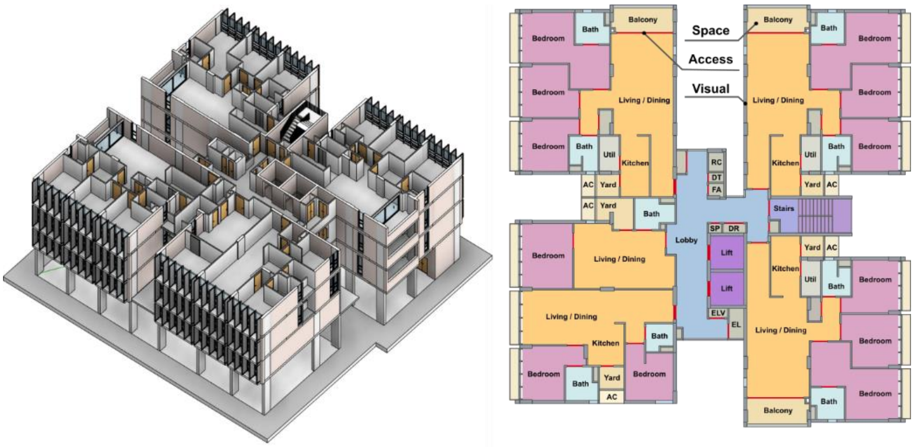


Figure 1. Residential building BIM (left) and generated regions (right).

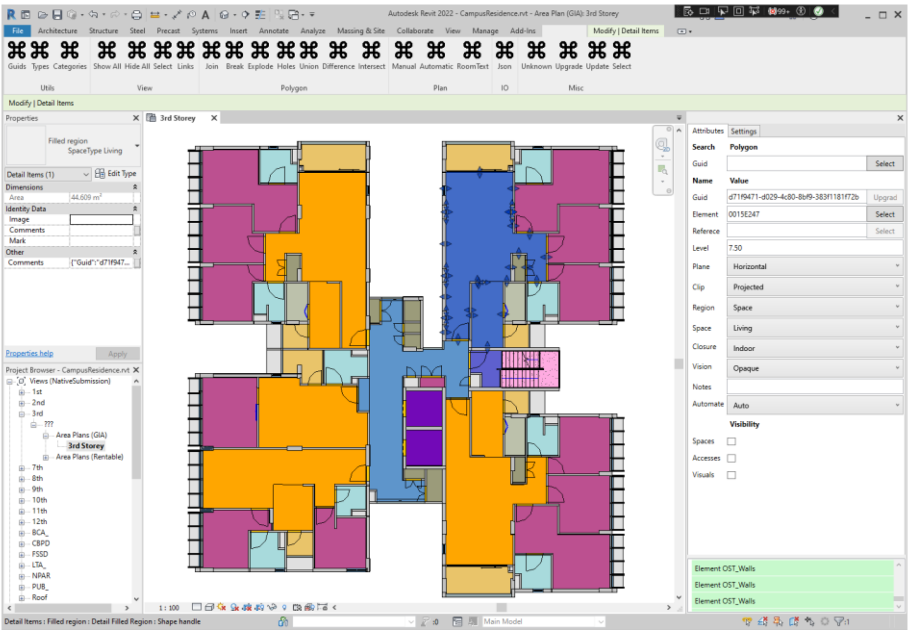
The process of generating regions from BIM is organized in three parts: (a) Element filtering and translation, where relevant entities are converted to regions; (b) Spatial reconstruction, where Boolean operations are performed among sets of regions to derive spaces; (c) Space identification, where space use characteristics are recovered; and (d) Graph generation, where spatial adjacencies are recovered. The data extraction tool was developed for Revit using its C#/.Net API. Offset and Boolean operations were performed using the Clipper2 library (Johnson, 2003) and space usage recovery was performed using the Quickenshtein library (Turner, 2020).

Overall, the computation reconstructs building plans, by performing intersections and projections between a reference plane and relevant building elements. Shape processing using two-dimensional entities was motivated by their computational efficiency when compared to volume reconstruction methods. In addition, as the resulting analysis visualizations are aimed at reporting for general audiences, the plan representation is also more meaningful. Nevertheless, while the primary output of those operations is planar, vertical information is not discarded but stored as region attributes.

2.1. ELEMENT FILTIERING AND TRANSLATION

BIM entities are strongly typed, in the sense that they follow as rigid object-model hierarchy, with their definition semantically associated with building element categories, such as, floors, walls, columns, doors and windows. This is the case with both the IFC standard and BIM software (Kereshmeh and Eastman, 2014). Nevertheless, due to conceptual limitations (Barekati et al. 2015; Belksy et al. 2016) certain ideas do not always map to concrete entity types (Tomaz and Ziga, 2008). For instance, there are no specialized entities for sidewalks and roads and generally urban features. This motivates the use of untyped generic models or standard floor elements with material properties or text annotations that hint about type. BIM entities are therefore essentially dynamically typed, for extensibility, because they allow arbitrary properties attachment, in a key-value pair sense. Moreover, there is a lack of industry standards and model quality control protocols, manifested by the difficulty of data normalization. Indicatively, there is no simple way to extract the opening width from a door because this is stored as a key-value pair where the key may be “width”, “rough width”, “clear width” or even using keys in various languages based on the manufacturer's origin. Finally, inconsistent use of geometric and semantic information is often encountered, such as elevators and parking lots, which are sometimes modelled as either floors, lines, hatches or bounding box entities. Therefore, extensive use of heuristics is required for filtering and classification.

The system first parses the well-defined entity types. Those are organized in groups: (a) Accessible regions are projections of entities including floors, ramps, flat roofs and topography; (b) Spatial boundary regions include walls, columns, facades, railings and parapets, which may be partially crossing or projected onto the reference plane; (c) Special building components such as elevators, plumbing, furniture, parking lots, planting which are used for semantic analysis and visualization. For each element group we define a set of extraction parameters which express the upper and lower bounding offsets from the base level plane used for clipping solids and deriving crossing and projected regions. For example, vertical circulation elements require clipping above and below the base level to capture transit between floors. Walls are clipped above the finished floor level to extract boundaries and account for openings such as windows and doors. Additionally, curtain walls, apart from the base offset they are also clipped at a small offset above the finished floor level to capture raised seals and the relevant transoms are projected, because otherwise the mullions and glazing produce unrealistic space boundaries.


Autodesk Revit 2022 - CampusResidence.rvt - Area Plan (GIA): 3rd Storey

Properties: Filled region SpaceType Living
Detail Items (1)
Dimensions
Area 44.609 m²
Identity Data
Image
Comments
Mark
Other
Comments {"Guid":"d71f947...

Attributes
Search Polygon
Guid
Name Value
Guid d71f9471-d029-4c80-8bf9-383f1181f72b
Element 0015E247
Referece
Level 7.50
Plane Horizontal
Clip Projected
Region Space
Space Living
Closure Indoor
Vision Opaque
Notes
Automate Auto
Visibility
Spaces ☐
Accesses ☐
Visuals ☐

Element OST_Walls
Element OST_Walls
Element OST_Walls

Detail Items : Filled region : Detail Filled Region : Shape handle

Figure 2. Software application developed for spatial information extraction.

2.2. SPATIAL RECONSTRUCTION

The objective of space recovery is to extract the boundary in the sense of the Net Internal Area (NIA) of building spaces. Unfortunately, while the concept of a room is conceptually supported by BIM standards and applications, it is limiting and often incorrectly modelled. For example, rooms may exclude, partially or completely include walls, often room tags are used to mark entire apartments within a residential building rather than individual rooms, room separation lines are often used inconsistently or to override the BIM logic, and finally the relationships between the space and its bounding elements is inconsistently tracked especially when room separators are used.

From the regions produced earlier we reconstruct spaces by performing a sequence of Boolean operations. An intermediate enclosure representation is computed for space bounding elements. This amounts to projection for walls without their associated doors and windows, the union of mullions, transoms and glazing for curtain walls, the projected union of their solids for parapets, railings, bollards etc. The union of spatial boundaries is computed resulting in a hierarchical structure of polygonal domains. The graph where each node represents a polygon ring, with each optionally containing additional nested rings, is then traversed outside-in. If we assume that all building elements have non-zero thickness, then the Jordan curve theorem allows us to unnest the hierarchy in spaces, should we alternate from exterior to interior at every other level of the tree structure. Thus, we arrive at a set of spatial domains which may contain holes, representing columns, voids, nested rooms, as well as hints as per whether they are indoor or outdoor. However, this is insufficient to completely determine the presence of spaces, especially urban spaces at the ground level, shafts, atria, voids, where the presence of slab or otherwise is important. Accessible regions per level are first united to produce a maximum available region which is Boolean differenced by the spatial boundary union to verify the presence of space in the sense of where people may be physically situated (Fig. 3).

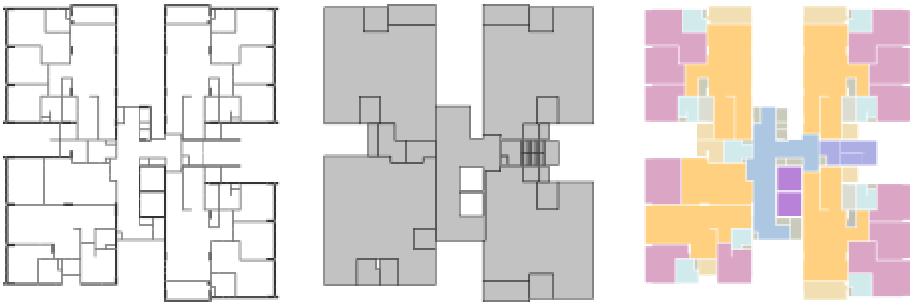
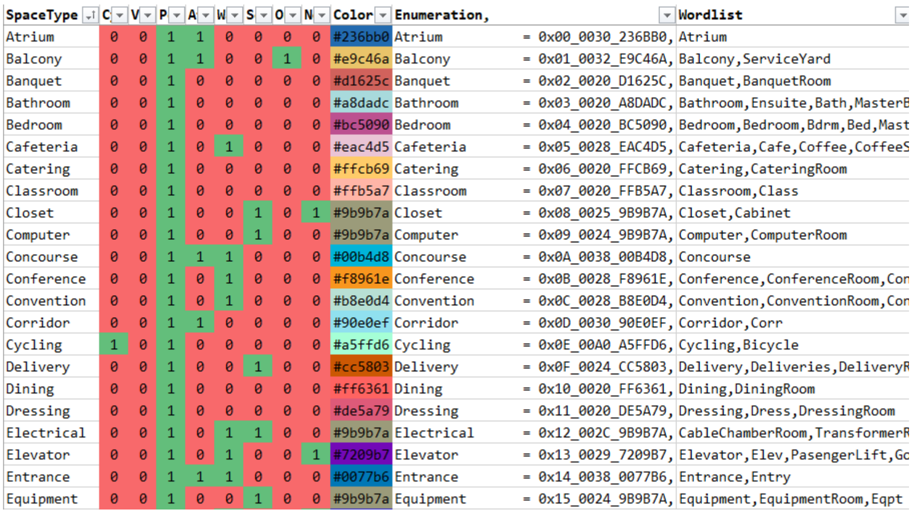


Figure 3. Space recovery using Boolean operations: Union of all spatial boundary regions (left). Union of all accessible regions (middle). Computed spatial regions (right).

Ideally, this procedure would suffice to compute all building spaces. However, we had to consider tolerances and modelling mistakes to improve the system's robustness. The most common problem with the models encountered is with walls that do not properly connect with one another causing gaps preventing enclosure or producing unions among adjacent spaces. This was approached using tolerances and heuristics. For instance, all spatial boundaries are dilated by a few centimetres to ensure overlap and thus spatial closure and separation. The offset is removed from the resulting spaces by erosion with the same tolerance. Accessible regions above the ground are artificially bounded with virtual walls. This ensures for instance that balconies are always accounted for, even if parapets or railings are missing, and such that spaces are always enclosed even if the slab edge does not meet curtain walls.

2.3. SPACE IDENTIFICATION

Spatial analysis, such as regulatory compliance checking, often requires information pertaining to the usage of each space. Therefore, spatial regions must be annotated automatically using BIM information available or inferred from the available semantic hints. While in theory a model's spatial element information may be sufficient, in practice it is rarely the case. The reason for this shortcoming is due to modelling practices, where rooms may be presents as either BIM entities or may appear using plain text annotations or even via linked CAD documents, and lack of standards in terms of a nomenclature and relevant abbreviations for space usage. We developed a list of typical space uses, such as "elevator", their known synonyms, such as "lift", "goods lift", "passenger lift" and abbreviations such as "elv". Space classes are also tagged as indoor or outdoor, primary, or secondary such as service rooms, and accessible by pedestrians, cyclists and/or vehicles (Fig. 4).



SpaceType	C	V	P	A	W	S	O	N	Color	Enumeration,	Wordlist
Atrium	0	0	1	1	0	0	0	0	#236bb0	Atrium = 0x00_0030_236BB0,	Atrium
Balcony	0	0	1	1	0	0	1	0	#e9c46a	Balcony = 0x01_0032_E9C46A,	Balcony,ServiceYard
Banquet	0	0	1	0	0	0	0	0	#d1625c	Banquet = 0x02_0020_D1625C,	Banquet,BanquetRoom
Bathroom	0	0	1	0	0	0	0	0	#a8dadc	Bathroom = 0x03_0020_A8DADC,	Bathroom,Ensuite,Bath,MasterB
Bedroom	0	0	1	0	0	0	0	0	#bc5090	Bedroom = 0x04_0020_BC5090,	Bedroom,Bedroom,Bdrm,Bed,Mast
Cafeteria	0	0	1	0	1	0	0	0	#eac4d5	Cafeteria = 0x05_0028_EAC4D5,	Cafeteria,Cafe,Coffee,CoffeeS
Catering	0	0	1	0	0	0	0	0	#ffcb69	Catering = 0x06_0020_FFCB69,	Catering,CateringRoom
Classroom	0	0	1	0	0	0	0	0	#ffb5a7	Classroom = 0x07_0020_FFB5A7,	Classroom,Class
Closet	0	0	1	0	0	1	0	1	#9b9b7a	Closet = 0x08_0025_9B9B7A,	Closet,Cabinet
Computer	0	0	1	0	0	1	0	0	#9b9b7a	Computer = 0x09_0024_9B9B7A,	Computer,ComputerRoom
Concourse	0	0	1	1	1	0	0	0	#00b4d8	Concourse = 0x0A_0038_00B4D8,	Concourse
Conference	0	0	1	0	1	0	0	0	#f8961e	Conference = 0x0B_0028_F8961E,	Conference,ConferenceRoom,Con
Convention	0	0	1	0	1	0	0	0	#b8e0d4	Convention = 0x0C_0028_B8E0D4,	Convention,ConventionRoom,Con
Corridor	0	0	1	1	0	0	0	0	#90e0ef	Corridor = 0x0D_0030_90E0EF,	Corridor,Corr
Cycling	1	0	1	0	0	0	0	0	#a5ffd6	Cycling = 0x0E_00A0_A5FFD6,	Cycling,Bicycle
Delivery	0	0	1	0	0	1	0	0	#cc5803	Delivery = 0x0F_0024_CC5803,	Delivery,Deliveries,DeliveryR
Dining	0	0	1	0	0	0	0	0	#ff6361	Dining = 0x10_0020_FF6361,	Dining,DiningRoom
Dressing	0	0	1	0	0	0	0	0	#de5a79	Dressing = 0x11_0020_DE5A79,	Dressing,Dress,DressingRoom
Electrical	0	0	1	0	1	1	0	0	#9b9b7a	Electrical = 0x12_002C_9B9B7A,	CableChamberRoom,TransformerR
Elevator	0	0	1	0	1	0	0	1	#7209b7	Elevator = 0x13_0029_7209B7,	Elevator,Elev,PasengerLift,Go
Entrance	0	0	1	1	1	0	0	0	#0077b6	Entrance = 0x14_0038_0077B6,	Entrance,Entry
Equipment	0	0	1	0	0	1	0	0	#9b9b7a	Equipment = 0x15_0024_9B9B7A,	Equipment,EquipmentRoom,Eqpt

Figure 4. Space typology, attributes, and keywords mapping.

To recover space usage from the model we perform a series of heuristics: (a) The geometric centre of every space is tested for containment against all available rooms, (b) the centre of each room is tested against each space, (c) presence of special features such as stairs, lifts and plumbing equipment is used for inferring stairwells, lift shafts, bathrooms and services, (d) material associations are used to infer roads, planting, water bodies etc. (e) plan area is used to label small spaces such as closets and services, (f) lack of access is used to label shafts and voids. Finally, to decipher arbitrary room names and abbreviations thereof, we employ the Levenshtein editing distance metric against the table of known types. Spaces that fail all of the tests are marked as unknown and require manual labelling using the user interface tools.

2.4. GRAPH GENERATION

Topological information is recovered from the spatial adjacencies by preforming point-in-polygon and point-on-segment queries. The process is computationally expensive therefore a bounding box hierarchy is used to accelerate the reconstruction. The result of this process is a logical graph with nodes representing spaces and access regions and edges their adjacency. Additional data collected during this process is stored within the polygons such that their points and segments are associated with spatial boundaries. The logical graph my be visualized for simple spatial arrangements, where spaces have convex shape, and their nodes are situated in their geometric centres. For complex spatial arrangements this is not possible as the graphs' edges overlap spatial boundaries. A polygon skeletonization procedure was thus employed, based on the Medial Axis Transform (Blum, 1967) shape analysis methodology (Lee, 2004), to generate spatially situated graphs that are contextually meaningful (Fig. 5).

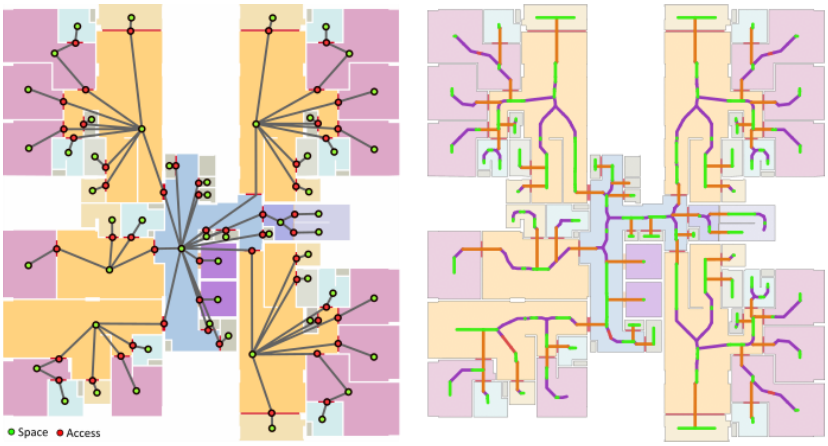


Figure 5. *Logical graph with nodes situated at the geometric middle of spaces and accesses (left) and graph generated using the Medial Axis Transform method (right).*

3. Results

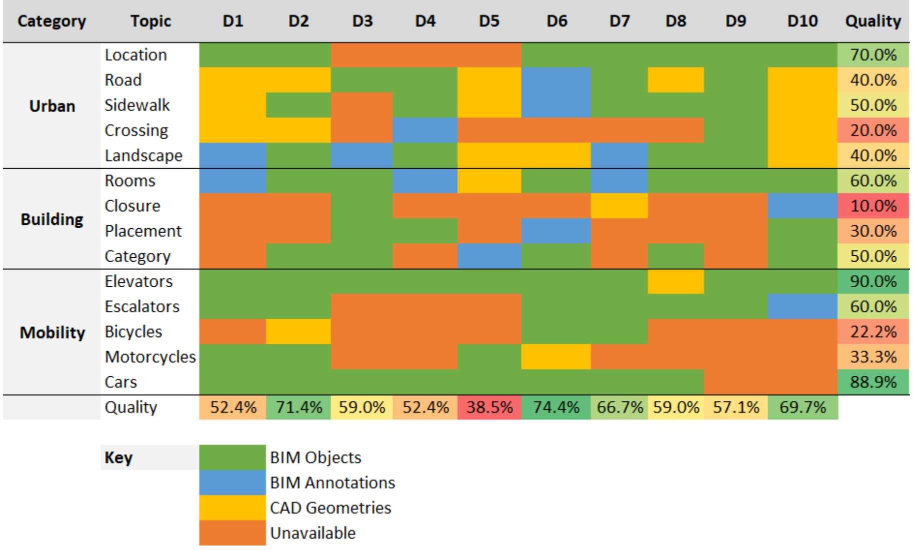
The system was developed using the university campus residential buildings as its initial case study. The campus residences BIM represents a benchmark or prototype of a properly modelled building which can be completely automatically analysed. There were no special procedures employed to build this model other than using the correct BIM entities for each purpose, ensuring that elements were properly hosted, walls met at their end points and rooms were all labelled. Subsequently, we validated the system using ten models submitted by private architecture firms for regulatory assessment to planning authorities. These include residential, commercial, transportation and mixed-use developments. The names and locations of the developments cannot be disclosed for confidentiality purposes and as such labelled as D1 to D10 (Fig. 6).

		D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	
Project Information	**Site**	28.6	11.6	36.4	14.7	10.9	6.2	9.8	18.6	17.0	39.2	x1000 m²
	GFA	71.5	82.4	64.1	47.9	38.2	22.2	28.4	67.0	4.9	168.7	x1000 m²
	Levels	14	17	12	18	26	10	11	19	3	7	
Polygon Analysis	**Generated**	526	431	805	184	280	307	251	724	231	1088	
	Relabelled	208	160	73	199	50	13	31	260	96	45	
	Drawn	816	86	655	431	387	125	182	454	258	701	
	Completed	1146	510	1320	687	428	427	499	733	474	1527	

Figure 6. *Project information including site area, gross floor area and number of floors for each of the ten developments analysed. Number of spatial polygons automatically generated and those that had to be manually relabelled and geometrically corrected.*

Overall, the tools developed were able to extract spatial structure from all projects but not without significant challenges requiring manual intervention or creating new heuristics. Corrections were performed for (a) malformed and incomplete geometries and (b) rooms with missing or incorrect tags were relabelled. The number of polygons manually adjusted for geometry corrections was 52% with standard deviation of 21% while relabelling was required for 16% of all polygons with standard deviation circa 12%. The rather pronounced magnitude of the standard deviation highlights the wide range of quality characteristics of the models which for geometry spans from 17% to 90% and for room labels from 3% to 35%. We did not identify correlations between the projects' sizes, in terms of gross floor area (GFA), and the number of geometry and metadata errors, with coefficients of -0.1 and -0.09, respectively.

The specific problems uncovered were explained earlier under the methods section and are presented below (Fig. 7). They are categorized under three semantic groups, namely (a) Urban: related to problems such as absence of project geolocation, properly modelled roads, sidewalks, pedestrian crossings and landscape features; (b) Building: related to proper annotation of rooms, closure of spaces, elements placement, and use of appropriate BIM element categories; (c) Mobility: related to presence of circulatory features such as escalators, elevators and vehicular information such as cyclist and motorist driving and parking related BIM elements. The average across all projects and categories are presented for highlighting the frequency and by association the severity of the problems. Geometric mistakes are more frequently encountered compared to semantic errors with only one in ten projects having properly enclosed spaces. We also note that while BIM is the primary medium of documentation there are still substantial amounts of information contained within associated CAD drawings. In summary, the sources of those problems may be attributed to (a) conceptual limitations of today’s BIM, (b) lack of digital industry standards and (c) lack of methods for model quality control. We observed significant challenges in spatial structure recovery at ground level of all buildings examined in contrast with upper floors. We attribute this to a conceptual hiatus between BIM and GIS, the building itself and its surrounding environment.



Category	Topic	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	Quality
Urban	Location	🟩	🟩	🟧	🟧	🟧	🟩	🟩	🟩	🟩	🟩	70.0%
	Road	🟨	🟨	🟩	🟩	🟨	🟦	🟩	🟨	🟩	🟨	40.0%
	Sidewalk	🟨	🟩	🟧	🟩	🟨	🟦	🟩	🟩	🟩	🟨	50.0%
	Crossing	🟨	🟨	🟧	🟦	🟧	🟧	🟧	🟧	🟩	🟨	20.0%
	Landscape	🟦	🟩	🟦	🟩	🟨	🟨	🟦	🟩	🟩	🟨	40.0%
Building	Rooms	🟦	🟩	🟩	🟦	🟨	🟩	🟦	🟩	🟩	🟩	60.0%
	Closure	🟧	🟧	🟩	🟧	🟧	🟧	🟨	🟧	🟧	🟦	10.0%
	Placement	🟧	🟧	🟩	🟩	🟧	🟦	🟧	🟧	🟧	🟩	30.0%
	Category	🟧	🟩	🟩	🟧	🟦	🟩	🟧	🟩	🟧	🟩	50.0%
Mobility	Elevators	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟩	🟩	90.0%
	Escalators	🟩	🟩	🟧	🟧	🟧	🟩	🟩	🟩	🟩	🟦	60.0%
	Bicycles	🟧	🟨	🟧	🟧	🟧	🟩	🟩	🟧	🟧	🟧	22.2%
	Motorcycles	🟩	🟩	🟧	🟧	🟩	🟨	🟧	🟧	🟧	🟧	33.3%
	Cars	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟧	🟧	88.9%
	Quality	52.4%	71.4%	59.0%	52.4%	38.5%	74.4%	66.7%	59.0%	57.1%	69.7%	

Key

🟩	BIM Objects
🟦	BIM Annotations
🟨	CAD Geometries
🟧	Unavailable

Figure 7. Summary of typical errors and overall model quality assessment per project and category.

4. Discussion

The challenges identified earlier motivated breaking the spatial structure recovery processes into separate tasks, namely (a) fully automated, where the entire model may be processed without user intervention, (b) quasi-automated, where elements are extracted under user-tuned settings and/or when automatically generated information are adjusted, such as updating semantic attributes, via the user-interface, and (c) manual extraction where regions are manually drawn in the CAD sense and their attributes are entered via the user interface.

The university residence model and the analysis tools were developed in parallel. Use of the spatial structure recovery tools during the development of the model in retrospect was instrumental in improving its quality and ensuring that no geometric or semantic issues were present in the final model. This is because periodic execution of the processes highlighted modelling mistakes such as geometric, topological, and semantic omissions and/or ambiguities. Therefore, we foresee that such tools that aim to comprehend BIM models may also be used for quality control as well as education.

A fundamental limitation of the tools presented is in that they primarily operate in two dimensions which may be insufficient for spatial analytical processes such as acoustics and environmental performance evaluation (Alam and Ham, 2014). Additionally, recovering information using heuristics resulted in a substantial number of parameters requiring careful calibration when there is a need for adjustment. Perhaps the use of machine learning may assist in overcoming this problem in the future (Koo et al. 2019; Ahmadpanah et al. 2023).

5. Conclusions

In conclusion, we presented a process for recovering spatial structure from BIM highlighting the challenges identified and the solutions developed. The investigation demonstrated the limitations of current BIM technologies and offers hints about how they may be improved in the future. Applications employing the information extracted from BIM, namely circulatory analysis and building code compliance is the subject of forthcoming work.

Acknowledgements

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Cities of Tomorrow R&D Programme (COT-H1-2020-2). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Ministry of National Development, Singapore.

References

Ahmadpanah H., Haidar A. and Latifi, SM.: 2023. BIM and Machine Learning (ML) Integration in Design Coordination: Using ML to automate object classification for clash detection. ECAADE, Graz, 619-628.

Alam J. and Ham J. J.: 2014, Towards A BIM-based energy rating system: Comparisons between FirstRate5 and ArchiCAD EcoDesigner. CAADRIA, Kyoto, 285-294.

Balaban O., Yağmur Kilimci, ES. Cagdas, G.: 2012. Automated Code Compliance Checking Model for Fire Egress Codes. ECAADE, Prague, 117-125.

Barekati E. Clayton M. J. and Yan, W.: 2015. A BIM-compatible schema for architectural programming information. CAAD Futures, Liege, 311-328.

Belsky M., Sacks R. and Brilakis I.: 2016. Semantic Enrichment for Building Information Modeling. Computer-Aided Civil and Infrastructure Engineering, 31, 261-274.

Blum H.: 1967. A transformation for extracting new descriptors of shape. Models for the Perception of Speech and Visual Form: 362-380.

Chen LJ. and Luo H.: 2014. A BIM-based construction quality management model and its applications. Automation in Construction, 46, 64-73.

Eastman C., Lee JM., Jeong YS., and Lee JK.: 2009. Automatic rule-based checking of building designs, Automation in Construction, 18(8), 1011-1033.

Eastman C. M., Jeong YS., Sacks R. and Kaner, I.: 2010. Exchange model and exchange object concepts for implementation of national BIM standard, Journal of Computing in Civil Engineering, 24(1), 25-34.

Eastman C., Teicholz P., Sacks R., and Liston K.: 2011. BIM handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors. Wiley.

Firas AD., Wei Y. and Edin J.: 2023. Campusim: An Integrated Parametric BIM for Campus Design Simulation and Optimization. CAADRIA, Ahmedabad, 471-480.

Howard R. and Andersen J. L.: 2001. Classification of building information - European and IT systems. ITcon. Mpumalunga, 9:1-14.

Johnson A.: 2003. Clipper2: A Polygon Clipping and Offsetting Library. Available at https://github.com/AngusJohnson/Clipper2

Kereshmeh A. and Eastman C.: 2014, Categorization of building product models in BIM Content Library portals, SIGRADI, Montevideo, 370-374.

Koo B., La S., Cho NW. and Yu Y.: 2019, Using support vector machines to classify building elements for checking the semantic integrity of building information models. Automation in Construction, 98, 183-194.

Lee J.: 2004. A spatial access oriented implementation of a 3-d GIS topological data model for urban entities. GeoInformatica 8 (3): 237-264.

Lee JK. Eastman C. Lee J. Kannala M. and Jeong YS.: 2010. Computing Walking Distances within Buildings Using the Universal Circulation Network. Environment and Planning B. 37(4), 628-645.

Migilinskasa D., Popovb V., Juoceviciusc V., and Ustinovichiusd L.: 2013. The Benefits, Obstacles and Problems of Practical Bim Implementation. Procedia Engineering, 57, 767-774.

Sherif A., Lee J. and Eastman C.: 2011. Automated Cost Analysis of Concept Design BIM Models. CAAD Futures, Liege, 403-418.

Tomaz P. and Ziga T.: 2008: Interoperability in practice: Geometric data exchange using the IFC standard. ITcon, 13, 362-380.

Turner J.: 2020. Quickenshtein: A quick and memory efficient Levenshtein Distance calculator for .NET. Available at: https://github.com/Turnerj/Quickenshtein

Wang J., Wang X., Shou W., Guo J., and Hou L.: 2014. Development of BIM Model Fitness Review System for Modelling Quality Control. Computing in Civil and Building Engineering, 577-584.