

IFC-based Algorithms for Automated Quantity Takeoff from Architectural Model: A Case Study on Residential Development Project

Temitope Akanbi, Ph.D., A.M.ASCE¹; Jiansong Zhang, Ph.D., A.M.ASCE^{2*}

Abstract

The estimation of construction quantities is a critical aspect of any construction project, and accuracy in the quantity takeoff (QTO) process is an integral part in achieving project success. Accuracy can be improved by automating the QTO process. However, one of the gaps in automated QTO is found to be the lack of a platform-ignorant computing algorithm with regard to different building information modeling (BIM) authoring tools. Available BIM-based commercial cost estimation software programs such as Autodesk Naviswork, Autodesk QTO, iTWO CostX, Innovaya Visual QTO, ConstrucSim, etc., are limited in terms of full automation, largely because full interoperability across all the different applications has not yet been achieved. This paper explores interoperable QTO algorithms using IFC-based BIM. To evaluate the IFC-based QTO algorithms, a case study was conducted on a residential development project in Kalamazoo, MI. Comparing with commercial software used in practice, the IFC-based QTO algorithms showed consistent results whereas more powerful with regard to its independence from BIM creation tools, in extracting the volumetric and areal quantities of building components. The IFC-based QTO algorithms processed the different types of building element configurations in the residential development project smoothly. Furthermore, the QTO

¹ Research Associate, Automation and Intelligent Construction Lab (AutoIC), School of Construction Management Technology, Purdue University, West Lafayette, IN. 47907; email: takanbi@purdue.edu

² Assistant Professor, Automation and Intelligent Construction Lab (AutoIC), School of Construction Management Technology, Purdue University, West Lafayette, IN. 47907; (* Corresponding Author) email: zhan3062@purdue.edu

algorithms are expected to reduce the manual input required in generating QTO through interfacing with IFC data which is the ISO international standard for BIM.

Keywords: Building Information Modeling (BIM); Quantity Takeoff (QTO); Case Study; Automation.

Introduction

Building Information Modeling (BIM)'s adoption/implementation in the architecture, engineering, and construction (AEC) industry is changing the way professionals and experts in the industry conduct their businesses (Feng et al. 2021; Godager et al. 2021). Currently BIM tools are utilized for design modeling (e.g., Revit, ArchiCAD, and SketchUp), costing (e.g., Navisworks, Timberline, DProfiler), scheduling (e.g., Synchro, Navisworks), collaboration and project delivery (e.g., Trimble Connect, Autodesk BIM 360, and Revizto), among others. According to Santos et al. (2017), these tools when fully exploited can provide immense benefits to owners and AEC professionals in various ways from conceptual design (e.g., improved coordination, clash detection, preconstruction project visualization) to operation/maintenance (e.g., as-built documentation, facility management, energy assessment). To take full advantages of all the benefits BIM tools can offer, IFC-based BIMs have been heavily researched to address the gaps in the development of these tools such as in the lack of interoperability, and in the lack of full automation of construction tasks such as QTO. However, problems still exist in the development of a system that both automates cost estimation and ensures seamless interoperability with other BIM tools. In achieving high accuracy, seamless data exchange between the different stakeholders' platforms is needed to avoid data transfer problems such as misrepresentation and missing data (Nawari 2012). The foundation for BIM to accomplish smooth data exchange lies in its interoperability (Cheung et al. 2012; Santos et al. 2017).

However, the lack of industry-wide interoperability has been plaguing the AEC industry since the embracement of BIM in the early 2000s (Akanbi and Zhang 2017). This interoperability gap led to the development of industry foundation classes (IFC). However, despite the establishment of such standards, full interoperability (seamless information exchange) between different BIM software programs is yet to be achieved. Gaps such as the lack of a proper exchange mechanism or software application that implements model-based interoperability still stand in the way (Sacks et al. 2010). For example, Cheung et al. (2012) identified an interoperability problem using IFC-based building information models (BIMs), namely, different tools store and keep information in proprietary formats which vary a lot in how IFC objects and properties are used. There are several researchers that have proposed methods to improve the interoperability of BIM-based tools. For example, Ahn et al. (2014) developed a BIM interface to enhance interoperability between CAD tools and dynamic energy analysis tools. Cemesova et al. (2015) proposed an extension to the IFC schema to aid the transfer of data and enhance interoperability between BIM and low energy design software. Choi et al. (2016) developed an openBIM-based environment to improve the interoperability of BIM-based energy performance assessments (EPAs). Karan et al. (2016) used semantic web technology to extend BIM interoperability to pre-construction tasks leveraging geospatial analysis. Ramaji and Memari (2018) developed the Interpreted Information Exchange (IIE) mechanism to automate the data exchange between architectural BIM and structural models. Kamel and Memari (2018) developed an Automated Building Energy Modeling and Assessment Tool (ABEMAT) to automate building energy modeling and simulation based on BIM, to make the process more reliable, accurate and faster comparing to the state of the art. In the realm of BIM-based QTO to support cost estimation, Alzraiee (2020) developed a Navisworks add-in and structured query language (SQL) for extracting needed information from BIM, Cepni and

Akcamete (2021) proposed a framework to use Revit Dynamo for material QTO from architectural BIM, there is a gap in in-depth investigation of QTO algorithms directly built upon IFC-based architectural BIM data (Alzraiee 2020; Cepni and Akcamete 2021). Rouhanizadeh et al. (2021) proposed an IFC-based cost estimation framework for transportation project which is along the line of addressing a similar gap in transportation projects, however, because of the fundamental difference between transportation projects and building projects especially their respective adoptions of BIM, the gap in IFC-based architectural BIM QTO persists. Major differences in the modeling of infrastructures (i.e., horizontal construction) when compared to the modeling of buildings (i.e., vertical construction) that need to be accounted for while applying BIM-based methods include: (1) difference in structure and components (e.g., vertical construction typically have openings whereas horizontal construction do not); (2) difference in vocabulary (e.g., piers in bridges as opposed to columns in buildings); and (3) differences in modeling procedures and techniques (Cheng et al. 2016). These differences make QTO procedures for infrastructure projects differ significantly from QTO for building projects.

To address this gap, in this paper, the authors conducted an in-depth case study for using IFC-based algorithms to support QTO from architectural model of a real residential development project. In this paper, the QTO algorithms were generated using a cutting-edge data-driven IFC-based algorithm development method, and the QTO algorithms focus on building projects (e.g., student apartment complex) rather than infrastructure projects. The IFC-based BIM QTO algorithms leverage the fundamental geometric representations of BIM objects to extract quantities of building components generated from any IFC-compliant BIM authoring tool, to support BIM interoperability better than the state-of-the-art commercial tools.

IFC-Based QTO Algorithms

Cost Estimation

Cost estimation in the AEC industry has always been a challenging and critical task. BIM-based cost estimation methods have helped semi-automate some of the processes thus reducing the time involved in cost estimation and improving the accuracy of cost estimates in comparison to traditional manual cost estimation. Despite these benefits provided by BIM-based tools in generating cost estimates, automation of the complete processes of construction cost estimation still requires further development. There are several issues and challenges posed by current methods, including: (1) interoperability issues – one-to-one interactions between the various software tools/platforms may cause errors/missing data; (2) labor intensiveness – manual efforts needed for the classification of model elements into different categories; (3) accuracy and level of development – the information extracted from the model is dependent on the quality of the input model; among others. The case study in this manuscript was limited to and focused on addressing gaps in BIM-based interoperability in generating QTO. Addressing all potential challenges and gaps in BIM-based cost estimation processes is not the intent of this case study.

Automated Quantity Takeoff (QTO)

Cost estimation is a very important aspect of any construction project and the accuracy in the cost estimate is necessary to the project's overall success (Lee et al. 2014; Akanbi and Zhang 2017). Currently, for the various commercial estimating software in use in the construction industry, the process(es) involved in achieving the estimates are similar. Most software requires a three-step operation to achieve cost estimation (Fig. 1). Step 1: Importing 3D models - an estimator imports a building design in BIM (or 3D drawings). Step 2: Taking off the quantities - the estimator classifies elements of the model into different categories (e.g., internal walls

mm partition), and the software executes the quantity computations. Step 3: Costing - the quantities generated are exported and linked to a cost database to assign costs to the various categories identified from Step 2. Two main challenges posed by these methods are the issues in collaboration (thus interoperability) and the needed human input. For the collaboration/interoperability issue, Step 1 and Step 3 of this general cost estimation process involve collaborative data exchanges with different BIM tools. Different tools store and keep information in different proprietary formats, which can impede the successful exchange of data. Without a reliable, complete, and successful data exchange, the reliability of cost estimates is questionable. For the second issue in human judgement needs, although QTO has been automated with the advent of BIM, manual efforts by human estimators are still required in classifying and matching the model elements to their various categories (Lee et al. 2014). This can introduce human errors and therefore affect cost estimation accuracy. BIM is revolutionizing the AEC industry by enabling a digital workflow and business process (Franco et al. 2015; Lu et al. 2013; Nassar 2012). BIM tools are comprehensive and are consisted of architectural design tools (e.g., Revit and ArchiCAD), structural design tools (e.g., Tekla and STAAD Pro), energy assessment tools (e.g., VE pro and Simuwatt), and cost estimation tools (e.g., DesignEst Pro and Autodesk Naviswork), among others (Cheung et al. 2012). Example benefits of BIM tools include design optimization, schedule optimization, project coordination support, and cost estimation automation. However, present BIM use was mostly on few purposes (e.g., 3D visualization, design support, clash detection) (Kreider et al. 2010). BIM use in other functions need further development to realize more potential benefits to building owners and industry professionals. One of the most useful such development is in automated QTO (Franco et al. 2015; Monteiro and Martins 2013; Plebankiewicz et al. 2015). As shown in the survey by Monteiro and Martins

(2013), BIM-based QTO can significantly improve the accuracy and efficiency in the cost estimation process. In as much as the study revealed, many efforts have utilized BIM tools to automate QTO. However, their survey results showed that automation of the complete estimating process (e.g., work item selection and costing) is still underexplored and requires further research/development. IFC-based BIMs have been heavily researched on to address: (1) construction interoperability issues, and (2) automation in construction management tasks such as cost estimation. However, in supporting cost estimation, problems still exist in the development of an intelligent system that not only automates QTO but enables seamless interoperability with other BIM software. To address these problems in BIM interoperability to support cost estimation, in this paper, the authors conducted a case study to evaluate the application of IFC-based QTO algorithms on a wood construction residential development project in Kalamazoo, MI. A wood project was chosen because of the prevalence of wood structure in the residential construction sector in the U.S., and according to Allen and Iano (2009), wood is the only major structural material that is renewable.

The focus on processing information using an IFC-based BIM improves interoperability across the different software tools upstream and downstream because: (1) IFC is an ISO standard for open, transparent, and neutral BIM data; and (2) almost all major BIM software support compatibility with IFC. In addition, the direct use of geometric representation information to the level of Cartesian points overcomes potential barriers brought by the different use of IFC schemas in different BIM tools and enables a robust interoperability. The IFC-based QTO algorithms can be directly applied to extracting quantities and material from building components in architectural BIM.

The automated QTO is performed by IFC-based algorithms that automatically retrieve the quantities of building elements and their associated material information needed in cost estimation through parsing the fundamental representations of building elements in the IFC-based BIM. The algorithm leveraged patterns in the IFC representations' structures and entities and attributes of the target building element's representation in the IFC data (Fig. 2).

Step 1: Model Development

A building information model is exported into the IFC format in this step. Many BIM authoring tools and BIM analytical applications have the compatibility with IFC data. For example, Autodesk, Bentley Systems, GRAPHISOFT, Solibri, and Trimble Tekla are some of the BIM authoring tools that are currently using the Coordination View (CV) to export and import BIM files and it is almost rare to see BIM applications without exportation option to IFC.

Step 2: Model View Definition (MVD)-based Checking

To address the interoperability gap in the AEC domain, this step validates the input BIM file using model view definition (MVD). With the validation process, the accuracy of the IFC-based BIM instance file is evaluated to check if it fulfils the specified exchange requirements. The MVD model is based on a subset of the existing MVD Coordination View Version 2.0 (CV V2.0), to check the input IFC file. The subset of the CV V2.0 used in creating the MVD model contains essential IFC entities that are used to represent information of the building components needed for QTO and pricing.

Step 3: Data Analysis

The information required in QTO are identified and analyzed (Figs. 3a & 3b) in this step. Fig. 3a shows the flowchart of the primary algorithm developed for wall QTO whereas Fig. 3b

shows the flowchart of the primary algorithm developed for floor QTO. As illustrated in Fig. 3a, *Process 1* temporarily reads all entities and their attributes into an array list to perform different extraction processes for the required variables. *Processes 2.1 to 3.2* extract the body representation (i.e., geometry) of a target wall instance. *Processes 4.1 to 10* analyze the body representation of the wall in two cases – using “SweptSolid” representation or using “Clipping” representation. *Processes 11.1 to 14* analyze the dimensions of potential opening elements in the wall. *Processes 15 to 17* analyze the material information of the wall. To illustrate the rationale used in the algorithms, the tracing pattern of a window opening (Fig. 4) was presented in detail as an example.

In the process chart shown in Fig. 3a, *Process 2.1* performs a search for “IFCWALLSTANDARDCase” entity and writes the entity into *WallStandardCase*. Then *Process 2.2* pulls out the 7th attribute of the *WallStandardCase* (an IFCPRODUCTDEFINITIONSHAPE) and writes this entity into *WallRepresentation*. Following that, *Process 3.1* pulls out the 2nd component in the 3rd attribute of the *WallRepresentation* (an IFCSHAPEREPRESENTATION) and writes this entity into *BodyRepresentationOne*. Then, *Process 3.2* pulls out the 1st component of the 3rd attribute of *WallRepresentation* and writes this entity into *BodyRepresentationTwo*. At this level, depending on the type of *BodyRepresentation* (i.e., “SweptSolid,” “Clipping,” “Brep,” or “SurfaceModel”), the corresponding case branch is activated for extracting the corresponding geometric parameters and calculating the quantities.

As an example of tracing entities and attributes to extract needed parameters for an opening (shown in Fig. 4), *Process 11.1* pulls out all lines containing IFCOPENINGELEMENT into *OpeningElement*. *Process 11.2* pulls out the 7th attribute of *OpeningElement* (an

IFCPRODUCTDEFINITIONSHAPE) and writes it into *OpeningRepresentation*. *Process 12* pulls out the 1st element of the 3rd attribute of *OpeningRepresentation* (an IFCSHAPEREPRESENTATION) and writes it into *OpeningDefShape*. *Process 13.1* pulls out the 1st element of the 4th attribute of *OpeningDefShape* (an IFCEXTRUDEDAREASOLID) and writes it into *OpeningItem*. *Process 13.2* extracts the 4th attribute of *OpeningItem* into *OpeningWidth*; *Process 13.2* further extracts the 1st attribute of *OpeningItem* (an IFCRECTANGLEPROFILEDEF) into *OpeningArea*. *Process 14* pulls out the 4th and 5th attributes of *OpeningArea* and writes them into *XODim* (i.e., the opening length) and *YODim* (i.e., the opening width), respectively. At this point the quantity measures for an opening element would be successfully extracted.

Case Study Description

To test the use of the IFC-based QTO algorithms, the case study was conducted on a residential development project in Kalamazoo, MI. The details of the project are presented below.

Location

The case study project selected in this paper was a residential development project located in Kalamazoo, MI. Kalamazoo is a city in the southwest region of the state of Michigan, approximately 241 kilometers away from two major cities, Chicago, and Detroit. The residential development project is organized into three buildings, buildings 305, 306 and 307; the buildings are three stories each. The location and orientation of the buildings were designed with respect to the local climate. For this case study, we selected building 305; Fig. 5 shows the site plan of the selected building (Design Plus Inc, 2012).

Building 305

Building 305 has a footage of 183.02 m² and provides 144 beds in a mixture of room configurations including single occupant's bedrooms in four-bed/four bath unit styles and two-bed/two bath unit styles. The building has 38 units made up of 4,172.46 m² on three floor levels; 32 two-bedroom units and 6 four-bedroom units. The two bedroom consists of "Deluxe" upgrades which offers bigger floor spaces in comparison to the standard two-bedroom units. All units have interior finishes including in-unit washer & dryers, hardwood floors, solid-core wood doors, individually controlled HVAC units, and nine-foot ceilings. The authors estimated the project cost of building 305 to be \$15,710,096 using building data cost of 2019 (RSMeans 2019). Table 1 outlines the unit style mix while Table 2 summarizes the square footages of each functional area per unit type.

Unit Types

Building 305 comprises of four different unit types: (1) 2 – bedroom unit; (2) 2 – bedroom deluxe unit; (3) 2x2 bedroom deluxe unit; and (4) 4 – bedroom unit (Design Plus Inc, 2012). Each bedroom unit is equipped with a bathroom and the kitchen spaces have drawers and storage spaces for each occupant. The following sub-sections describe each of the four units in details:

2 – Bedroom Unit

In the 2 – bedroom unit, the space is divided into two separate bedrooms on opposite ends of the rooms flanking the shared spaces, that is, the living area and the kitchen space. The bedroom spaces are either 14.4 m² or 15.70 m² in size and are large enough to accommodate a single bed with a nightstand, a dresser, a desk, a chair, and a closet (Design Plus Inc, 2012). The

shared spaces are joined, and it is approximately 31.87 m² in size (Design Plus Inc, 2012). Fig. 6 shows the 2 – bedroom unit plan with dimensions.

2 – Bedroom Unit Deluxe

The 2-bedroom deluxe unit is similar in design to the standard 2-bedroom unit but offers more square footage and the space configuration is slightly different. The space is divided into two separate bedroom spaces clustered together at one end of the apartment with the shared spaces on the other end of the apartment. The bedroom spaces are either 14.68 m² or 18.12 m² in size while the joined shared space is 43.66 m² in size (Design Plus Inc, 2012). Fig. 7 shows the 2 – bedroom deluxe unit plan with dimensions.

2x2 Bedroom Deluxe Unit

The 2x2 bedroom deluxe unit is designed as a duplex comprising of two standard 2-bedroom units. The sizes of the functional spaces and space arrangements are similar to that of the standard 2-bedroom unit (Fig. 6). Fig. 8 shows the 2x2 bedroom deluxe unit plan with dimensions.

4 - Bedroom Deluxe Unit

In the 4 – bedroom unit, the space is divided into two bedrooms clusters flanking the shared spaces. The bedroom spaces are 14.86 m² in size while the joined shared space is approximately 46.92 m² in size (Design Plus Inc, 2012). Fig. 9 shows the 4 – bedroom unit plan with dimensions.

3D Model Development

Building 305 was not originally available in a 3D model, thus the original 2D design drawings were utilized to generate the 3D models used in this study. The models were developed by industry experts and BIM tools utilized for generating the 3D models were Autodesk Revit, Graphisoft ArchiCAD, and Trimble SketchUp. Figs. 10a, 10b, and 10c show the developed BIM instance models of building 305 in Revit (Model *A*), ArchiCAD (Model *B*), and SketchUp (Model *C*) respectively. Fig. 11 shows the exterior components of building 305 which comprises of CMU walls with brick veneer, roof shingles and aluminum windows/doors. The interior finishes are made up of carpet floor coverings, sheet vinyl floor coverings (kitchens and bathrooms), painted walls, and painted wood bases. For each BIM instance model of building 305 (Model *A*, *B* & *C*), the four different unit types were modeled and converted into IFC format, which serves as the input data for our QTO algorithms. Figs. 12a, 12b, 12c & 12d show the developed Revit BIM instance models of the 2 bedroom unit (Model *A*₁), 2 bedroom deluxe unit (Model *A*₂), 2x2 bedroom deluxe unit (Model *A*₃), and the 4 bedroom unit (Model *A*₄) respectively. Figs. 13a, 13b, 13c & 13d show the developed ArchiCAD BIM instance models of the 2 bedroom unit (Model *B*₁), 2 bedroom deluxe unit (Model *B*₂), 2x2 bedroom deluxe unit (Model *B*₃), and the 4 bedroom unit (Model *B*₄) respectively. Figs. 14a, 14b, 14c & 14d show the developed SketchUp BIM instance models of the 2 bedroom unit (Model *C*₁), 2 bedroom deluxe unit (Model *C*₂), 2x2 bedroom deluxe unit (Model *C*₃), and the 4 bedroom unit (Model *C*₄) respectively.

Evaluation

For evaluation of the IFC-based QTO algorithms' performance, through the case study, comparative analysis was conducted as detailed below.

Data

Twelve models (Figs. 12 – 14) were used for the evaluation of the QTO algorithms. The models were created based on the residential development project. Paper-based plans and drawings were acquired based upon which BIMs of the various unit styles were developed in three different BIM authoring tools - Revit, ArchiCAD, and SketchUp. These building information models (Figs. 12a – d, 13a – d, and 14a – d) were then further exported into IFC format which served as the input file for the QTO algorithms.

QTO Algorithms

Figs. 12, 13 & 14 present the BIMs of the twelve models used for evaluation. The input files (IFC file) for each individual unit were imported into the implemented QTO algorithms in python. The IFC-based QTO algorithms were then utilized in automatically performing the QTO for the building elements in the twelve models created using the three different BIM authoring tools. An example of the output result using the developed algorithms is displayed in Fig. 15. Fig. 15 shows the partial experimental result of the 2-bedroom unit (Model A_1) using the developed algorithms. In Fig. 15, “Index” represents the ID of the elements; “depth” is height of the wall element; “Xdim” is the wall length; “Ydim” is the wall width; “trim_one, trim_two, radius” are attributes of a curved wall utilized in calculating the length of a curved wall (length_of_wall); “openingwidth, xodim, yodim” are all quantities associated with openings and represent the width of an opening, the length of an opening, and the width of an opening, respectively. The quantities were rounded to three digits after the decimal point. The partial QTO results of the wall, floor, and roof components for models A_1 , B_2 , & C_3 were tabulated in Table 3.

Quantity Takeoff (QTO) Using Commercial Software

To compare with the state of the art, two BIM estimation tools, Software A, and Software B were utilized in this experiment. The models generated (Models $A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4, C_1, C_2, C_3$ & C_4) were saved in the proprietary files of each of the commercial estimating software. As an example, for Software A (in *format a*); the data in *format a* was then imported into the Software A platform. This process was repeated for Software B. Next the authors attempted to use the two commercial estimation tools to extract the QTO of all twelve building models in their respective native file formats.

Evaluation of the IFC-based QTO Algorithms

To evaluate the accuracy of the quantities generated by the IFC-based QTO algorithms, the authors compared the quantities obtained for Models ($A_1 - D_1$) with quantities generated using the commercial software.

Comparison of the Results

For the comparison of the results, the quantities of the wall, floor, and roof components of each unit type and configurations (Models $A_1 - C_3$) using the commercial software were compared with the quantities that were automatically generated using the IFC-based QTO algorithms. The results were recorded in a tabular format. Table 4 shows the partial results for the comparison of the quantities generated for six models in the three different BIM tools. All quantities were rounded to three digits after the decimal point to aid in the comparison. The results generated using the developed algorithms were consistent with the results from the commercial quantification tools. However, the comparison revealed that while the developed algorithms extracted the quantities from all models, the commercial estimating tools were not robust enough

to support direct access of each of the different input formats (*format a, format b, and format c*). Fig. 16 shows a bar chart that illustrates the comparison of the results in terms of successfully extracted quantities in each method. A comparison of robustness and comprehensiveness of the IFC-based QTO algorithms to the other two commercial estimation tools is summarized in Table 4 and Table 5. These are state-of-the-art estimation tools in the market by major BIM vendors. The comparison was mainly in checking what quantities could be successfully taken off in each tool. The results from Tables 4 and 5 suggest that although all three methods produced consistent QTO results; the commercial tools are limited in their capability to support different BIM authoring tools. In contrast, the IFC-based QTO algorithms successfully took off the quantities created in all the BIM authoring tools used. For other BIM authoring tools not tested in this experiment, as long as they are compatible with IFC (they are likely so), our QTO algorithms are expected to work as well, whereas there is no guarantee for the current proprietary estimation tools to work.

Contributions to the Body of Knowledge

This case study contributes to the body of knowledge in the following ways. First, despite extensive research leveraging BIM in QTO, our literature review shows existing efforts heavily relied on proprietary BIM tools such as Navisworks and Revit Dynamo. This case study demonstrated the use of IFC-based QTO algorithms directly for QTO from architectural BIM of a residential development project. In comparison with commercial proprietary software used in practice, such algorithms were shown to achieve equivalent quantities but were independent from BIM creation tools and therefore can support BIM interoperability better. In addition, the IFC-based QTO algorithms were shown to be able to process different unit types and configurations in the residential development project with success. Extension to other types of projects (e.g.,

industrial, healthcare) is within reach which will be tested in future work. Given sufficiently large dataset for training/development, the IFC-based QTO algorithms can be extended to cover different patterns in IFC entities/attributes usage including different geometric representation methods (e.g., SweptSolid, Brep, SurfaceModel, Clipping, CSG).

Practical Applications

The QTO algorithms tested in this paper can be utilized in extracting quantities of components designed in various BIM authoring tools. The algorithms were tested on three models developed from three different BIM platforms and the performance of the tested QTO algorithms were compared to that of the state-of-the-art commercial estimating software platforms. It shows that the state-of-the-art estimating tools are less robust than the developed QTO algorithms in extracting quantities from models developed in the different BIM workflows.

Limitations and Future Work

Five main limitations are acknowledged: (1) this case study only focuses on the QTO step of cost estimation, and the quality of the quantities generated depend on the quality of the input BIM. The implications of the presented method on other estimating issues such as duplication, omitting items, or relevant and realistic costs are out of the scope of this paper and can be investigated in future work; (2) this case study only focuses on building construction. While the same interoperability issue also presents itself in highway construction, its magnitude could be different from that in building construction. Such implications will also need to be studied in future work; (3) in this case study the tests were only conducted on explicitly modeled objects such as openings, staircases, and walls, objects not explicitly modeled (e.g., accessories) were not covered. In the practical use of IFC models, objects are not always represented explicitly (e.g., they may use IfcBuildingElementProxy instead of IfcWallStandardCase to represent a

wall). In order to process inexplicitly represented objects, automated BIM object classification will be needed. There is a rich set of existing methods in literature such as machine learning-based (e.g., Koo & Shin 2018; Koo et al. 2019) and rule-based methods (e.g., Sacks et al. 2017; Ma et al. 2018; Wu & Zhang 2020) on this currently-still-active research area of BIM object classification. The authors plan to test the integration of such methods into their IFC-based QTO workflow in future work; (4) quantities generated by the QTO algorithms are limited to the level of development of the input data (model), e.g., intersections between elements were determined based on their inherent object separation and representations (e.g., the geometric representations of walls, floors, columns, beams, etc.) in the input IFC models; (5) in this case study, we focused on a model that explicitly contains the necessary information required for QTO, as filtered by MVD. For unmodeled elements, models that do not contain the necessary information for QTO, or for QTO that cannot be achieved using existing IFC models, the authors plan to investigate it in their future work by extending the current QTO algorithm, such as through invariant signature-based and logic-based automated information inference as demonstrated by Wu and Zhang (2022) in automated building code compliance checking. Similar approach will also be tested in deriving higher level of details than what's already explicitly modeled in BIM.

Conclusions

As identified by several researchers, there is a gap in the development of an automated/expert system for cost estimation support that not only reduces/eliminates the need of manual efforts but also addresses the interoperability issue that is plaguing the construction industry. To address this gap, the authors conducted a case study to evaluate IFC-based QTO algorithms that facilitates QTO from architectural BIM generated from different BIM authoring

tools. The IFC-based QTO algorithms extracted the quantities needed for cost estimation by parsing the representations of the building elements in IFC.

The algorithms were evaluated on a residential development project in Kalamazoo, MI. The developed algorithms successfully extracted the quantities of the wall and floor components of the different building unit types and configurations in the project. The extracted results were compared to that from commercial software used in practice. It was found that consistent quantities were obtained whereas the authors' algorithms were more independent from BIM authoring tools, therefore providing better BIM interoperability support.

Data Availability Statement

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgement

The authors would like to thank the National Science Foundation (NSF). This material is based upon work supported by NSF under Grant No. 1745374. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

References

- Ahn, K-U., Kim, Y-J., Park, C-S., Kim, I., & Lee, K. (2014). "BIM interface for full vs. semi-automated building energy simulation" *Energy and Buildings.*, 68(2014), 671-678
- Akanbi, T., & Zhang, J. (2017). "Automated wood construction cost estimation." *Proc., ASCE International Workshop on Computing in Civil Engineering 2017*, ASCE, Reston, VA., 141-148.

- 417 Allen, E., and Iano, J. (2009). "Wood." *Fundamentals of Building Construction: Materials and*
418 *Methods*, 5th Ed., John Wiley & Sons Inc., Hoboken, New Jersey, 90.
- 419 Alzraiee, H. (2020). "Cost estimate system using structured query language in BIM."
420 *International Journal of Construction Management*, DOI:
421 10.1080/15623599.2020.1823061
- 422 Cemesova, A., Hopfe, C., & Mcleod, R. (2015). "PassivBIM: Enhancing interoperability between
423 BIM and low energy software." *Automation in Construction*, 57(2015), 17-32.
- 424 Cepni, Y., & Akcamete, A. (2021). "Enhancing BIM-based QTO using visual programing." *2021*
425 *European Conference on Computing in Construction*, Online eConference. < [https://ec-](https://ec-3.org/publications/conferences/2021/paper/?id=164)
426 [3.org/publications/conferences/2021/paper/?id=164](https://ec-3.org/publications/conferences/2021/paper/?id=164)> (Feb) 8th, 2022).
- 427 Cheung, F.K., Rihan, J., Tah, J., Duce, D., & Kurul, E. (2012). "Early stage multi-level cost
428 estimation for schematic BIM models." *Automation in Construction*, 27, 67-77.
- 429 Choi, J., Shin, J., Kim, M., & Kim, I. (2016). "Development of openBIM-based energy analysis
430 software to improve the interoperability of energy performance assessment." *Automation*
431 *in Construction*, 72(2016), 52-64.
- 432 Design Plus Inc., 2016. Western Mich. University Western View PH. 2.
- 433 Feng, Y., Wang, J., Fan, H., & Hu, Y. (2021). "A BIM-Based Coordination Support System for
434 Emergency Response." *IEEE Access*, 9(2021), 68814-68825,
435 10.1109/ACCESS.2021.3077237.
- 436 Franco, J., Mahdi, F., & Abaza, H. (2015). "Using Building Information Modeling (BIM) for
437 estimating and scheduling, adopting barriers." *Universal Journal of Management*,
438 9(2015), 376-384.

- 439 Godager, B., Onstein, E., & Huang, L. (2021). "A BIM-Based Coordination Support System for
440 Emergency Response." *IEEE Access*, 9(2021), 68814-68825,
441 10.1109/ACCESS.2021.3077237.
- 442 Kamel, E., & Memari, A.M. (2018). "Automated building energy modeling and assessment tool
443 (ABEMAT)." *Energy*, 147(2018), 15-24.
- 444 Karan, P., Irizarry, J., & Haymaker, J. (2016). "BIM and GIS integration and interoperability
445 based on semantic web technology." *J. Comput. Civ. Eng.*, 2016, 30(3): 04015043.
- 446 Koo, B., & Shin, B. (2018). "Applying novelty detection to identify model element to IFC class
447 misclassifications on architectural and infrastructure building information models." *J.*
448 *Comput. Des. Eng.*, 5 (4): 391–400.
- 449 Koo, B., La, S., Cho, N.W., & Yu, Y. (2019). "Using support vector machines to classify building
450 elements for checking the semantic integrity of building information models." *Autom.*
451 *Constr.*, 98 (Feb): 183–194.
- 452 Kreider, R., Messner, J., & Dubler, C. (2010). "Determining the frequency and impact of applying
453 BIM for different purposes on projects." <[http://bim.psu.edu/uses/Freq-](http://bim.psu.edu/uses/Freq-Benefit/BIM_Use-2010_Innovation_in_AEC-Kreider_Messner_Dubler.pdf)
454 [Benefit/BIM_Use-2010_Innovation_in_AEC-Kreider_Messner_Dubler.pdf](http://bim.psu.edu/uses/Freq-Benefit/BIM_Use-2010_Innovation_in_AEC-Kreider_Messner_Dubler.pdf) > (Sept.
455 24th, 2017).
- 456 Lee, S., Kim, K., & Yu, J. (2014). "BIM and ontology-based approach for building cost
457 estimation." *Automation in Construction*, 41, 96-105.
- 458 Lu, W., Peng, Y., Shen, Q., & Li, H. (2013). "Generic model for measuring benefits of BIM as a
459 learning tool in construction tasks." *Journal of Construction Engineering and*
460 *Management*, 139(2), 10.1061/(ASCE)CO.1943-7862.0000585.

- Ma, L., Sacks, R., Kattel, U., & Bloch, T. (2018). "3D object classification using geometric features and pairwise relationships." *Comput. -Aided Civ. Infrastruct. Eng.*, 33 (2): 152–164.
- Monteiro, A., & Martins, J. (2015). "A survey on modeling guidelines for quantity takeoff-oriented BIM-based design." *Automation in Construction*, 35(2013), 238-253.
- Nawari, N. (2012). "BIM standardization and wood structures." *Proc., ASCE International Conference on Computing in Civil Engineering*, ASCE, Reston, VA, 293-300.
- Plebankiewicz, E., Zima, K., & Skibniewski, M. (2015). "Analysis of the first polish BIM-Based cost estimation application." *Procedia Engineering*, 123(2015), 405-414.
- Ramaji, I.J., & Memari, A.M. (2018). "Interpretation of structural analytical models from the coordination view in building information models." *Automation in Construction*, 90 (Jun), 117–133.
- Rouhanizadeh, B., Kermanshachi, S., Ramaji, I.J., & Shakerian, S. (2021). "Development of an automated tool for cost estimation of transportation projects." *Proc., International Conference on Transportation and Development 2021*, ASCE, Reston, VA, 178-190.
- Sacks, R., Kaner, I., Eastman, M., & Jeong, Y.S. (2010). "The Rosewood experiment – building information modeling and interoperability for architectural precast facades." *Automation in Construction*, 19(2010), 419-432.
- Sacks, R., Ma, L., Yosef, R., & Borrmann, A. (2017). "Semantic enrichment for building information modeling: Procedure for compiling inference rules and operators for complex geometry." *J. Comput. Civ. Eng.*, 31 (6): 04017062.

- Santos, A., Costa, A., & Grillo, A. (2017). "Bibliometric analysis and review of Building Information Modeling literature published between 2005 and 2015." *Automation in Construction*, 80(2017), 118-136.
- Wu, J., & Zhang, J. (2022). "Model validation using invariant signatures and logic-based inference for automated building code compliance checking." *Journal of Computing in Civil Engineering*, 36(3), 04022002.
- Wu, J., & Zhang, J. (2019). "New automated BIM object classification method to support BIM interoperability." *J. Comput. Civ. Eng.*, 33(5), 04019033.
- Yu, W., Lai, C., & Lee, W. (2006). "A WICE approach to real-time construction cost estimation." *Automation in Construction*, 15, 12-19.

Final Accepted Version

Figure Captions List

Fig. 1. Typical Steps in Cost Estimation.

Fig. 2. IFC-based QTO Method.

Fig. 3a. Flowchart for the Primary Algorithm of Wall QTO.

Fig. 3b. Flowchart for the Primary Algorithm of Floor QTO.

Fig. 4. Tracing Pattern of an Opening.

Fig. 5. Architectural Site Plan -Building 305 (Design Plus Inc, 2012).

Fig. 6. Architectural Plans of the 2-Bedroom Unit (Design Plus Inc, 2012).

Fig. 7. Architectural Plans of the 2-Bedroom Deluxe Unit (Design Plus Inc, 2012).

Fig. 8. Architectural Plans of the 2x2 Bedroom Deluxe Unit (Design Plus Inc, 2012).

Fig. 9. Architectural Plans of the 4 - Bedroom Deluxe Unit (Design Plus Inc, 2012).

Fig. 10. BIM Instance Models of Building 305: (a) Model A in Autodesk Revit, (b) Model B in Graphisoft ArchiCAD, (c) Model C in Trimble SketchUp.

Fig. 11. Front View of the 3D Model Rendering of Building 305.

Fig. 12. BIM Instance Models in Autodesk Revit: (a) 2 Bedroom Unit (Model A₁), (b) 2 Bedroom Deluxe Unit (Model A₂), (c) 2x2 Bedroom Deluxe Unit (Model A₃), (d) 4 Bedroom Unit (Model A₄).

Fig. 13. BIM Instance Models in Graphisoft ArchiCAD: (a) 2 Bedroom Unit (Model B₁), (b) 2 Bedroom Deluxe Unit (Model B₂), (c) 2x2 Bedroom Deluxe Unit (Model B₃), (d) 4 Bedroom Unit (Model B₄).

Fig. 14. BIM Instance Models in Trimble SketchUp: (a) 2 Bedroom Unit (Model C₁), (b) 2 Bedroom Deluxe Unit (Model C₂), (c) 2x2 Bedroom Deluxe Unit (Model C₃), (d) 4 Bedroom Unit (Model C₄).

Fig. 15. Partial Experimental Results of a 2-bedroom Unit (Model A₁) using the Developed QTO Algorithms.

Fig. 16. Comparison of the Results Using the Developed Algorithms and Two State-of-the-Art Commercial Tools.

Table 1. Unit mix

Apartment Style	Count	m ²	Percentage Mix
2 – Bedroom	24	1,797.12	63
2 – Bedroom Deluxe	4	359.72	11
2 X 2 Bedroom Deluxe	4	668.90	11
4 – Bedroom	6	797.67	15

Table 2. Square footages breakdown (Building 305)

Areas	2- Bedroom Unit (m ²)	2- Bedroom Deluxe Unit (m ²)	2x2 Bedroom Deluxe Unit (m ²)	4- Bedroom Unit (m ²)	1st Floor (m ²)	2nd Floor (m ²)	3rd Floor (m ²)
Bedroom 1 w/ closet	15.70	14.68	29.73	14.86			
Bedroom 2 w/ closet	14.4	18.12	28.89	14.86			
Bedroom 3 w/ closet				15.05			
Bedroom 4 w/ closet				14.86			
Bathroom 1	5.57	5.02	11.61	5.57			
Bathroom 2	5.57	5.39	12.82	6.04			
Bathroom 3				5.57			
Bathroom 4				6.04			
Living Room w/ Kitchen	31.87	43.66	77.57	46.92			
Linen	0.84	1.21	2.60	1.11			
Laundry	0.93	1.86	3.99	2.04			
Corridor					55.74	55.74	55.74
Stairs					65.03	65.03	65.03
Misc.					62.25	62.25	62.25
Unit Area (m ²)	74.88	89.93	167.23	132.94			
Total Units	2.23	0.37	0.37	0.56			
Total Area (m ²)	1797.12	359.72	668.90	797.67	183.02	183.02	183.02
Total Building Area (m ³)							4172.47

Table 3. QTO results of selected building objects

Model	Component	Length (m)	Width (m)	Height (m)	Area (m ²)	Volume (m ³)
<i>A₁</i>	Wall 1	3.248	0.305	3.048	9.900	3.019
	Wall 2	0.610	0.305	3.048	1.859	0.567
	Wall 3	0.914	0.305	3.048	2.786	0.850
	Wall 4	7.344	0.200	3.048	22.385	4.477
	Wall 5	4.518	0.089	3.048	13.771	1.226
	Floor	10.734	7.935	0.579	85.174	49.316
		Length (m)	Width (m)	Area (m ²)	Volume (m ³)	Slope (°)
	Roof 1	9.296	0.051	85.179	4.327	30.000
	Component	Length (m)	Width (m)	Height (m)	Area (m ²)	Volume (m ³)
<i>B₂</i>	Wall 1	3.962	0.305	3.048	12.076	3.683
	Wall 2	0.457	0.305	3.048	1.393	0.425
	Wall 3	0.914	0.305	3.048	2.786	0.850
	Wall 4	6.096	0.200	3.048	18.581	3.716
	Wall 5	5.029	0.089	3.048	15.328	1.364
	Floor	10.738	7.115	0.579	76.401	44.236
		Length (m)	Width (m)	Area (m ²)	Volume (m ³)	Slope (°)
	Roof 1	9.300	0.051	76.401	3.881	30.000
	Component	Length (m)	Width (m)	Height (m)	Area (m ²)	Volume (m ³)
<i>C₃</i>	Wall 1	3.962	0.305	3.048	12.076	3.683
	Wall 2	0.610	0.305	3.048	1.859	0.567
	Wall 3	0.610	0.305	3.048	1.859	0.567
	Wall 4	3.962	0.305	3.048	12.076	3.683
	Wall 5	0.914	0.305	3.048	2.786	0.850
	Floor			0.579	18.293	4.865
		Length (m)	Width (m)	Area (m ²)	Volume (m ³)	Slope (°)
	Roof 1	7.728	3.298	25.487	47.217	-

Table 4. Comparison of experimental results

Model	Method	Component	Length (m)	Width (m)	Height (m)	Area (m ²)	Volume (m ³)
<i>A₁</i>	Developed Algorithm	Wall 1	3.248	0.305	3.048	9.900	3.019
	Software A		3.248	0.305	3.048	9.900	3.019
	Software B		-	-	-	-	-
	Developed Algorithm	Wall 2	0.610	0.305	3.048	1.859	0.567
	Software A		0.610	0.305	3.048	1.859	0.567
	Software B		-	-	-	-	-
	Developed Algorithm	Wall 3	0.914	0.305	3.048	2.786	0.850
	Software A		0.914	0.305	3.048	2.786	0.850
	Software B		-	-	-	-	-
	Developed Algorithm	Wall 4	7.344	0.200	3.048	22.385	4.477
	Software A		7.344	0.200	3.048	22.385	4.477
	Software B		-	-	-	-	-

B₁	<i>Developed Algorithm</i>	Floor	10.734	7.935	0.579	85.174	49.316
	<i>Software A</i>		10.734	7.935	0.579	85.174	49.316
	<i>Software B</i>		-	-	-	-	-
	<i>Developed Algorithm</i>	Wall 1	3.248	0.305	3.048	9.900	3.019
	<i>Software A</i>		-	-	-	-	-
	<i>Software B</i>		3.248	0.305	3.048	9.900	3.019
	<i>Developed Algorithm</i>	Wall 2	0.610	0.305	3.048	1.859	0.567
	<i>Software A</i>		-	-	-	-	-
	<i>Software B</i>		0.610	0.305	3.048	1.859	0.567
	<i>Developed Algorithm</i>	Wall 3	0.914	0.305	3.048	2.786	0.850
	<i>Software A</i>		-	-	-	-	-
	<i>Software B</i>		0.914	0.305	3.048	2.786	0.850
	<i>Developed Algorithm</i>	Wall 4	7.344	0.200	3.048	22.385	4.477
	<i>Software A</i>		-	-	-	-	-
	<i>Software B</i>		7.344	0.200	3.048	22.385	4.477
C₁	<i>Developed Algorithm</i>	Floor	10.734	7.935	0.579	85.174	49.316
	<i>Software A</i>		-	-	-	-	-
	<i>Software B</i>		10.734	7.935	0.579	85.174	49.316
	<i>Developed Algorithm</i>	Wall 1	3.248	0.305	3.048	9.900	3.019
	<i>Software A</i>		3.248	0.305	3.048	9.900	3.019
	<i>Software B</i>		-	-	-	-	-
	<i>Developed Algorithm</i>	Wall 2	0.610	0.305	3.048	1.859	0.567
	<i>Software A</i>		0.610	0.305	3.048	1.859	0.567
	<i>Software B</i>		-	-	-	-	-
	<i>Developed Algorithm</i>	Wall 3	0.914	0.305	3.048	2.786	0.850
	<i>Software A</i>		0.914	0.305	3.048	2.786	0.850
	<i>Software B</i>		-	-	-	-	-
	<i>Developed Algorithm</i>	Wall 4	7.344	0.200	3.048	22.385	4.477
	<i>Software A</i>		7.344	0.200	3.048	22.385	4.477
	<i>Software B</i>		-	-	-	-	-
A₂	<i>Developed Algorithm</i>	Floor	10.734	7.935	0.579	85.174	49.316
	<i>Software A</i>		10.734	7.935	0.579	85.174	49.316
	<i>Software B</i>		-	-	-	-	-
	<i>Developed Algorithm</i>	Wall 1	3.962	0.305	3.048	12.076	3.683
	<i>Software A</i>		3.962	0.305	3.048	12.076	3.683
	<i>Software B</i>		-	-	-	-	-
	<i>Developed Algorithm</i>	Wall 2	0.457	0.305	3.048	1.393	0.425
	<i>Software A</i>		0.457	0.305	3.048	1.393	0.425
	<i>Software B</i>		-	-	-	-	-
	<i>Developed Algorithm</i>	Wall 3	0.914	0.305	3.048	2.786	0.850

B₂	<i>Software A</i>		0.914	0.305	3.048	2.786	0.850
	<i>Software B</i>						
	<i>Developed Algorithm</i>	Wall 4	6.096	0.200	3.048	18.581	3.716
	<i>Software A</i>		6.096	0.200	3.048	18.581	3.716
	<i>Software B</i>						
	<i>Developed Algorithm</i>	Floor	10.738	7.115	0.579	76.401	44.236
	<i>Software A</i>		10.738	7.115	0.579	76.401	44.236
	<i>Software B</i>						
	<i>Developed Algorithm</i>	Wall 1	3.962	0.305	3.048	12.076	3.683
	<i>Software A</i>		3.962	0.305	3.048	12.076	3.683
	<i>Software B</i>						
	<i>Developed Algorithm</i>	Wall 2	0.457	0.305	3.048	1.393	3.683
	<i>Software A</i>		0.457	0.305	3.048	1.393	3.683
	<i>Software B</i>						
C₂	<i>Developed Algorithm</i>	Wall 3	0.914	0.305	3.048	2.786	0.850
	<i>Software A</i>		0.914	0.305	3.048	2.786	0.850
	<i>Software B</i>						
	<i>Developed Algorithm</i>	Wall 4	6.096	0.200	3.048	18.581	3.716
	<i>Software A</i>		6.096	0.200	3.048	18.581	3.716
	<i>Software B</i>						
	<i>Developed Algorithm</i>	Floor	10.738	7.115	0.579	76.401	44.236
	<i>Software A</i>		10.738	7.115	0.579	76.401	44.236
	<i>Software B</i>						
	<i>Developed Algorithm</i>	Wall 1	3.962	0.305	3.048	12.076	3.683
	<i>Software A</i>		3.962	0.305	3.048	12.076	3.683
	<i>Software B</i>						
	<i>Developed Algorithm</i>	Wall 2	0.457	0.305	3.048	1.393	3.683
	<i>Software A</i>		0.457	0.305	3.048	1.393	3.683
	<i>Software B</i>						

540

541

Table 5. Robustness evaluation

BIM tool	IFC-based QTO algorithm	Software A	Software B
Autodesk Revit	Yes	Yes	No
GraphiSOFT ArchiCAD	Yes	No	No
Trimble SketchUp	Yes	Yes	Yes
Other BIM Tools	Yes	?	?

Final Accepted Version

Figure 1

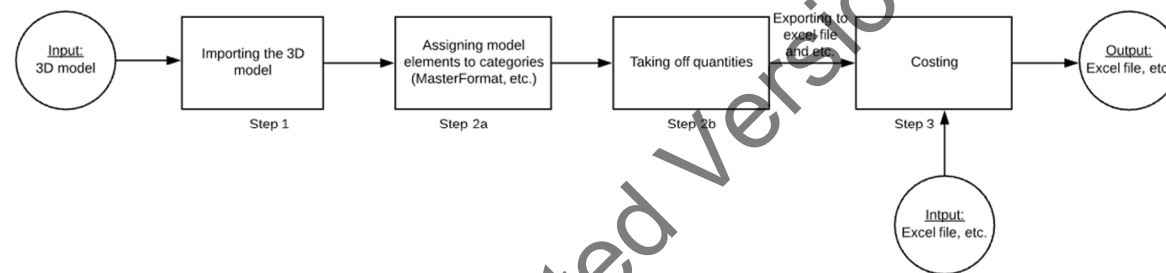
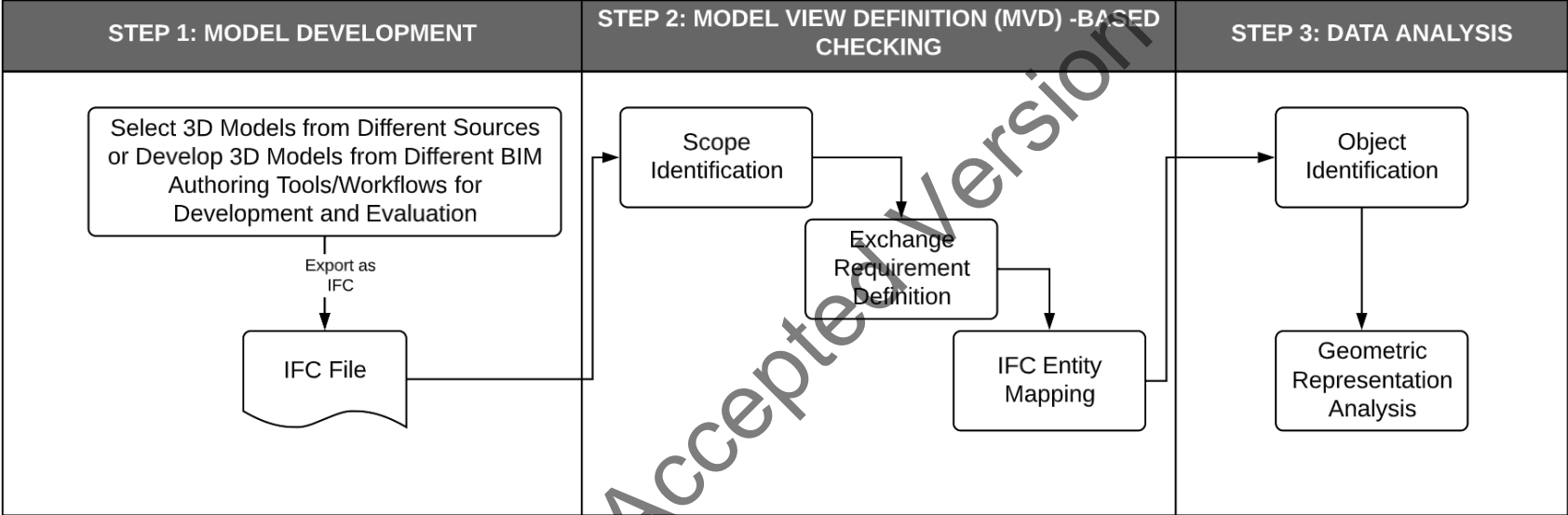
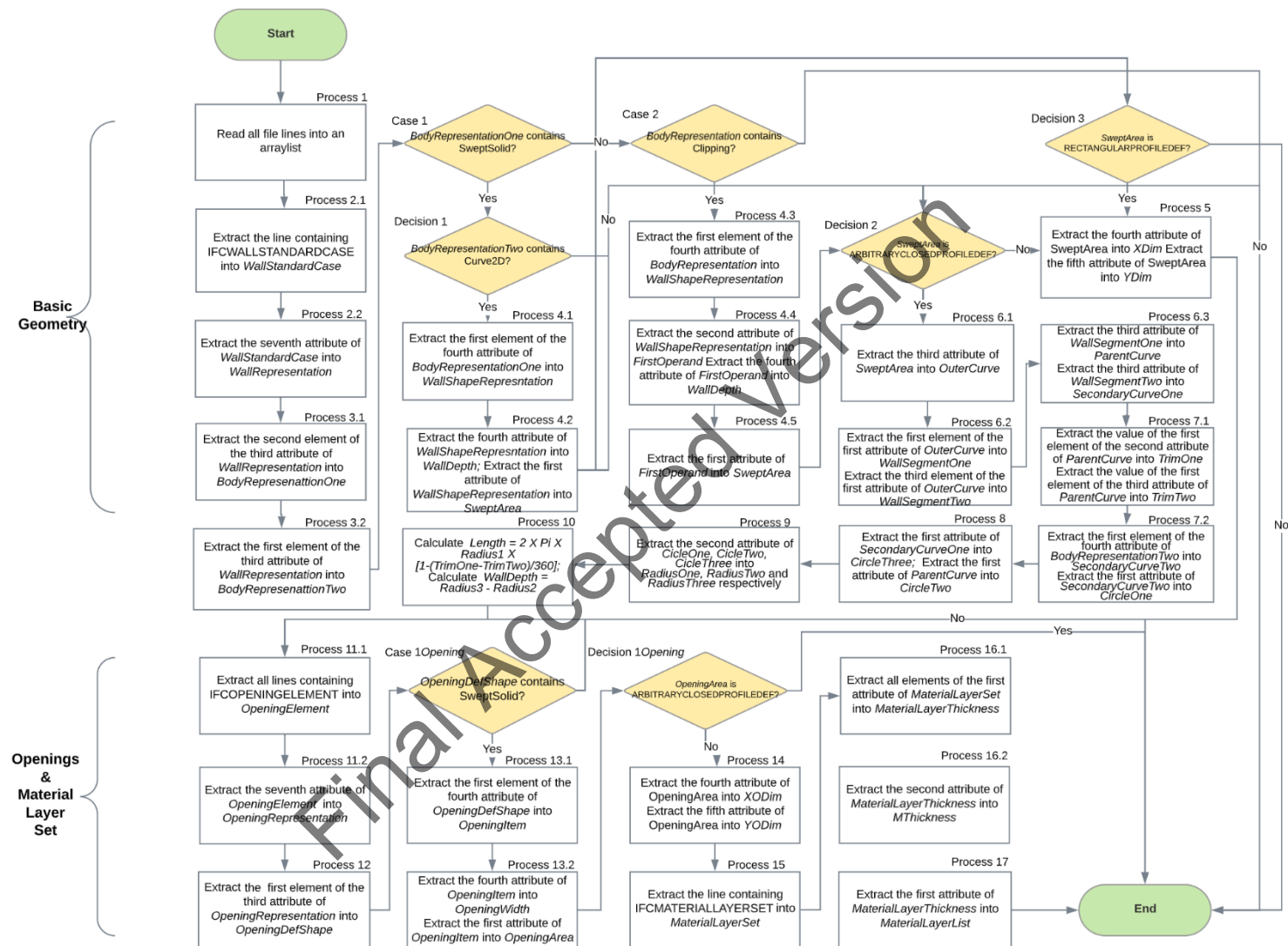
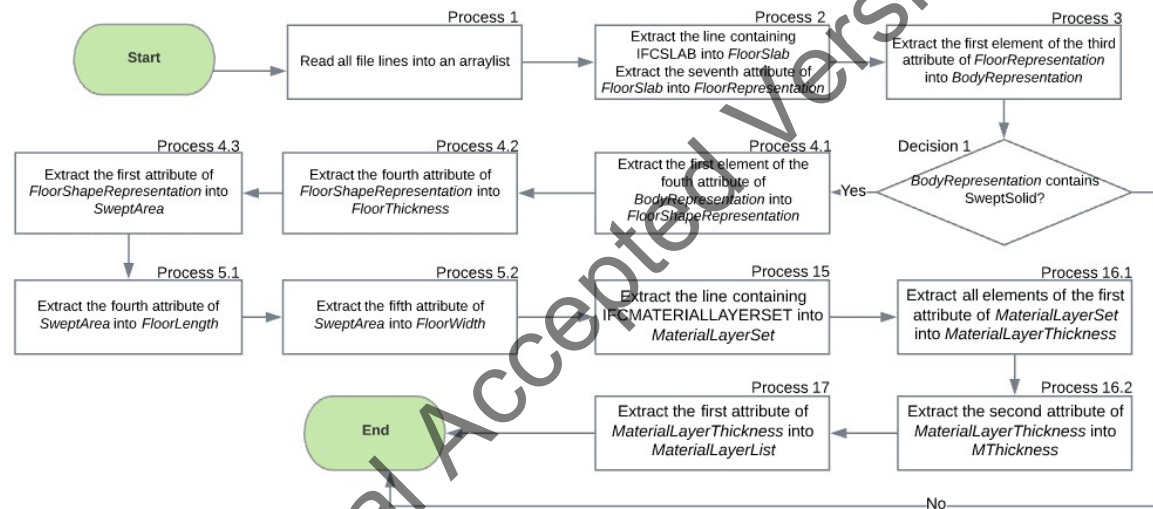
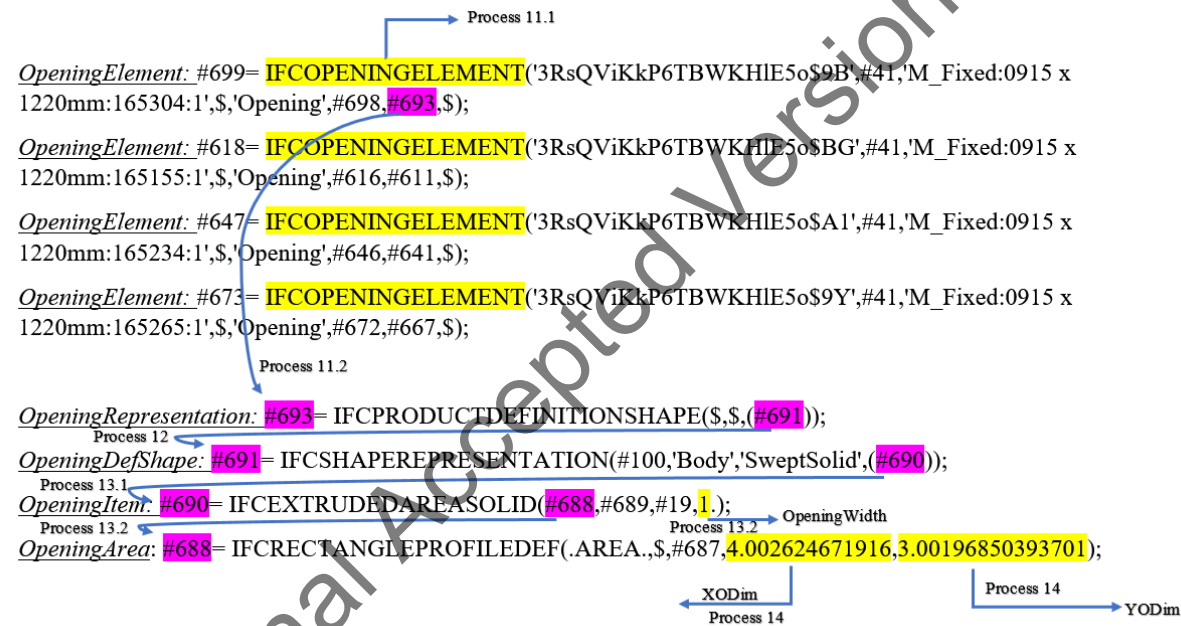


Figure 2









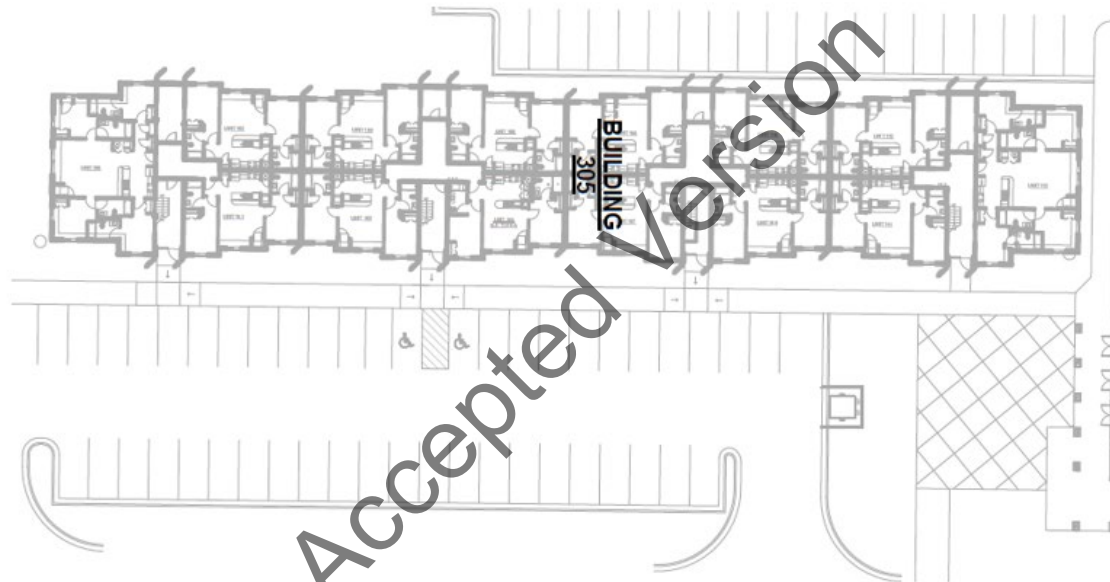
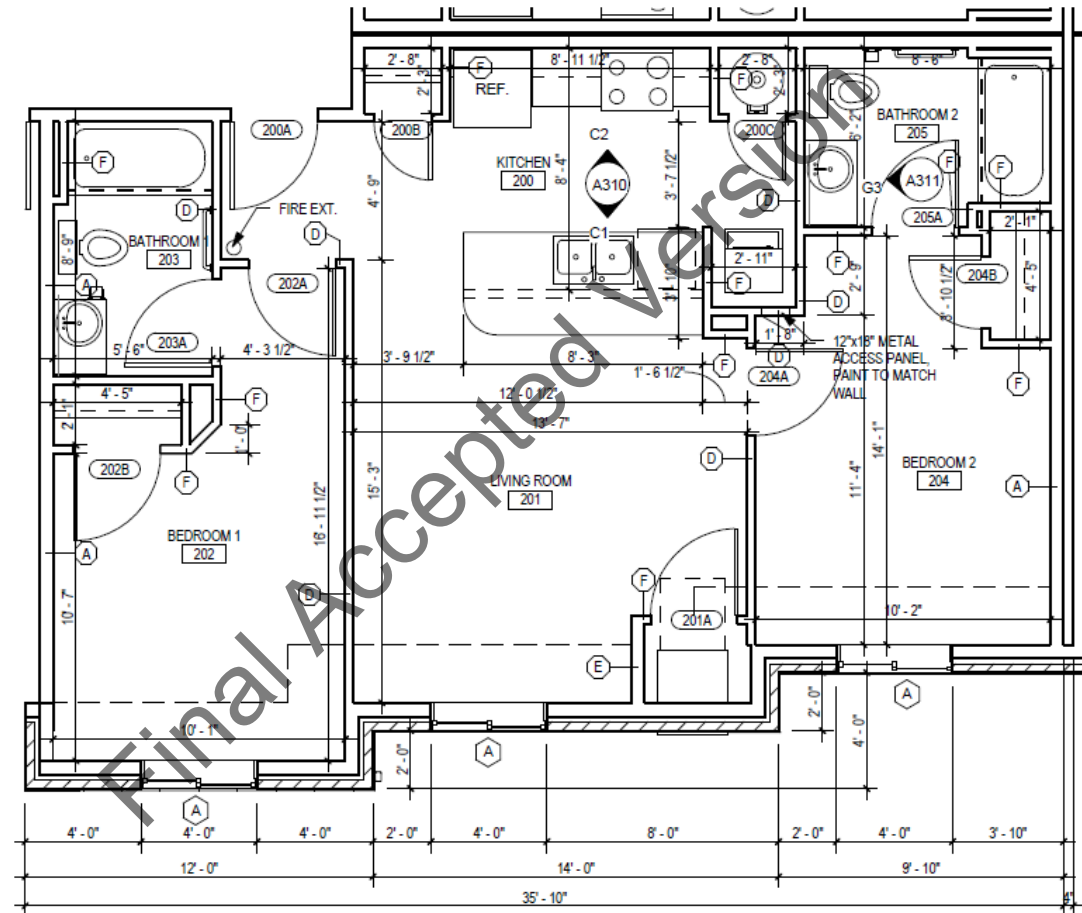
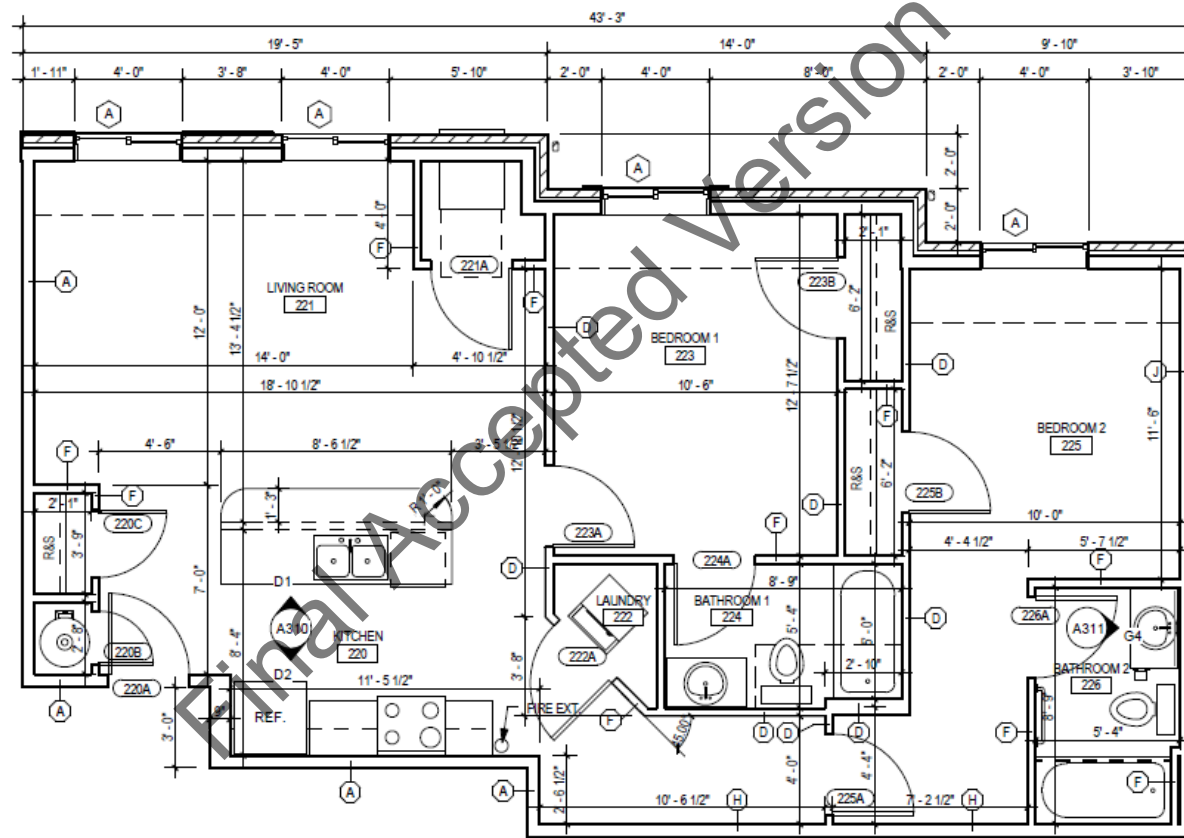
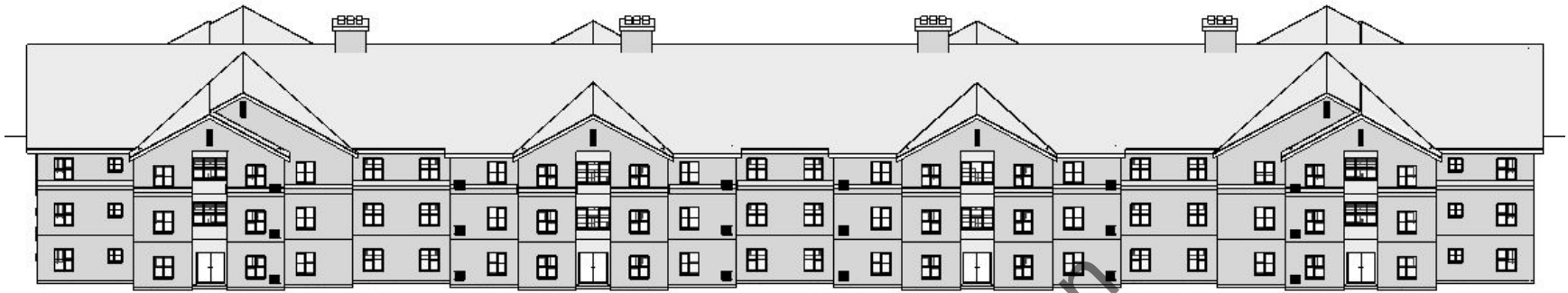


Figure 6





[illegible]



(a). Model A in Autodesk Revit



(b). Model B in Graphisoft ArchiCAD

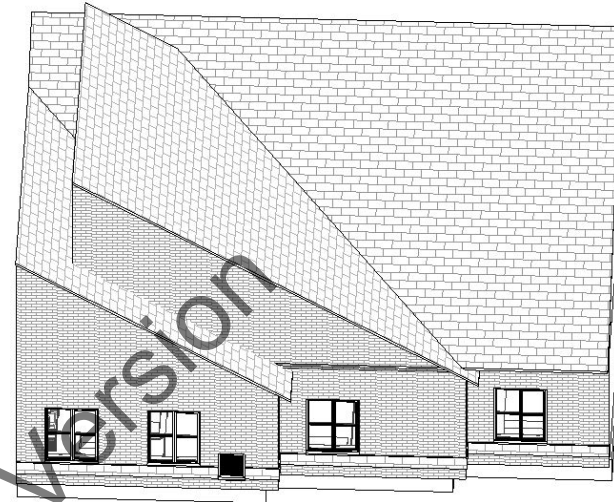


(c). Model C in Trimble SketchUp

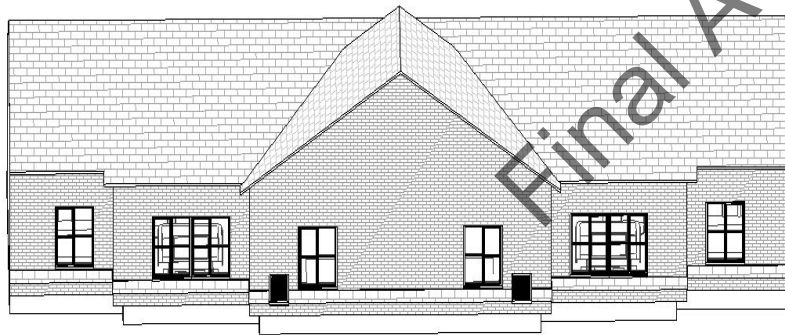




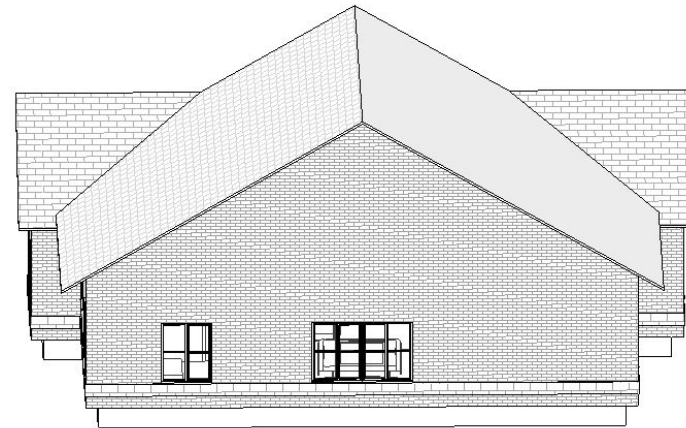
(a). 2 Bedroom Unit (Model A_1)



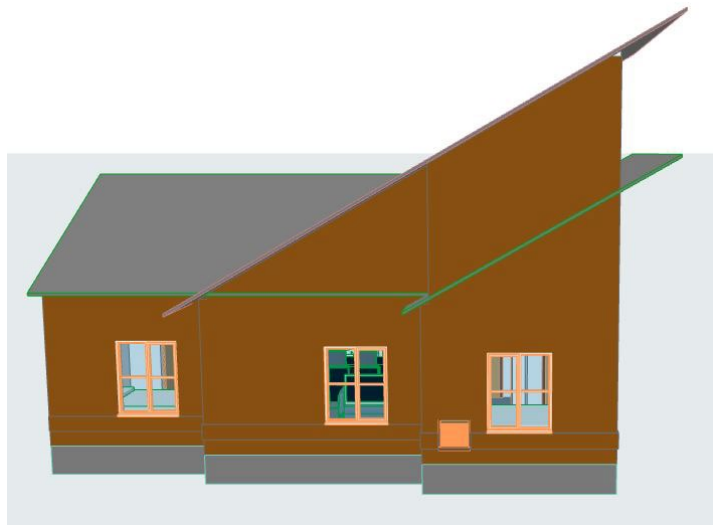
(b). 2 Bedroom Deluxe Unit (Model A_2)



(c). 2x2 Bedroom Deluxe Unit (Model A_3)



(d). 4 Bedroom Unit (Model A_4)



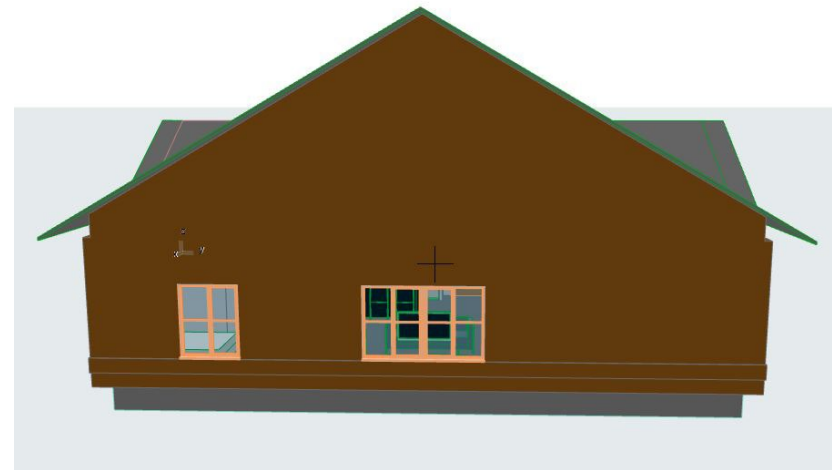
(a). 2 Bedroom Unit (Model B_1)



(b). 2 Bedroom Deluxe Unit (Model B_2)



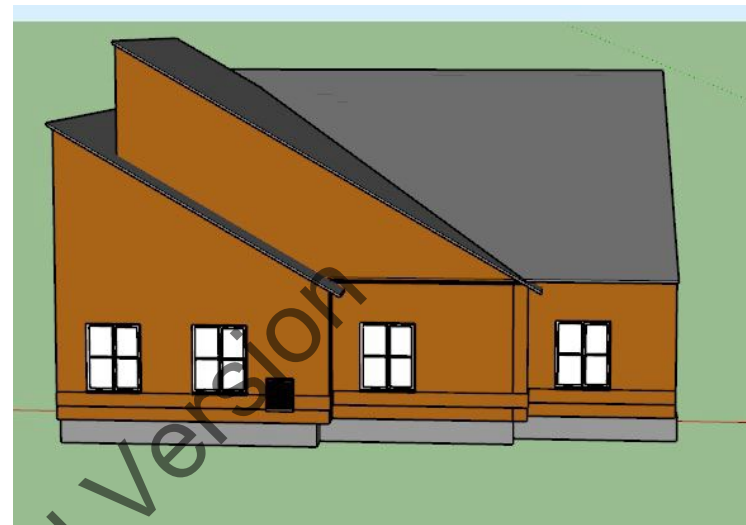
(c). 2x2 Bedroom Deluxe Unit (Model B_3)



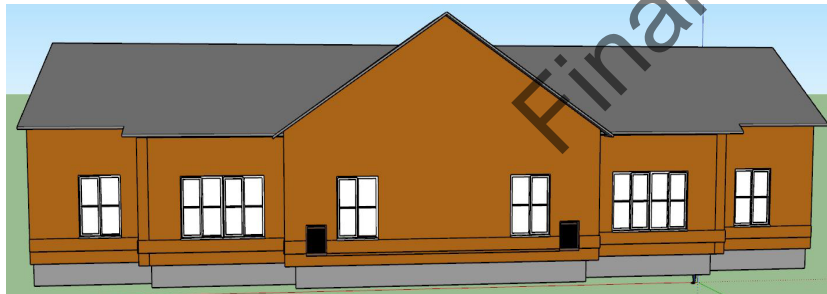
(d). 4 Bedroom Unit (Model B_4)



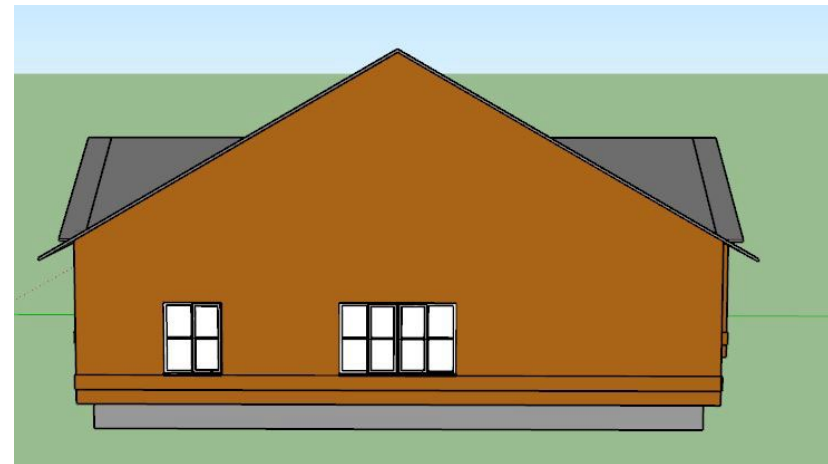
(a). 2 Bedroom Unit (Model C_1)



(b). 2 Bedroom Deluxe Unit (Model C_2)



(c). 2x2 Bedroom Deluxe Unit (Model C_3)



(d). 4 Bedroom Unit (Model C_4)

Index	depth	Xdim	Ydim	trim_one	trim_two	radius	openingWidth	xodim	yodim	length_of_wall
'1aNi6QhnX2hht76aIW1EOt'	10.	10.656167979...	1.	nan	nan	nan	nan	nan	nan	nan
'1aNi6QhnX2hht76aIW1EOo'	10.	2.0000000000...	1.	nan	nan	nan	nan	nan	nan	nan
'1aNi6QhnX2hht76aIW1EOh'	10.	2.9999999999...	1.	nan	nan	nan	nan	nan	nan	nan
'1aNi6QhnX2hht76aIW1ESN'	10.	24.095472440...	0.6561679790...	nan	nan	nan	nan	nan	nan	nan
'1aNi6QhnX2hht76aIW1ESJ'	10.	14.821850393...	0.2916666666...	nan	nan	nan	nan	nan	nan	nan
'1aNi6QhnX2hht76aIW1ESI'	10.	4.2916666666...	0.2916666666...	nan	nan	nan	nan	nan	nan	nan
'1aNi6QhnX2hht76aIW1ESc'	9.9999999999...	2.0616797900...	0.2624671916...	nan	nan	nan	nan	nan	nan	nan
'1aNi6QhnX2hht76aIW1ESb'	9.9999999999...	0.5	0.2624671916...	nan	nan	nan	nan	nan	nan	nan
'1aNi6QhnX2hht76aIW1ESW'	10.	11.135498687...	0.2916666666...	nan	nan	nan	nan	nan	nan	nan
'1aNi6QhnX2hht76aIW1ESI'	10.	4.2986376171...	0.2916666666...	nan	nan	nan	nan	nan	nan	nan
'1aNi6QhnX2hht76aIW1ESk'	10.	2.9948952575...	0.6561679790...	nan	nan	nan	nan	nan	nan	nan
'1aNi6QhnX2hht76bUW1ESy'	nan	nan	nan	nan	nan	nan	0.2916666666...	2.5	7.	nan
'3GOxdVHiJDwufvIyJ83zSm'	nan	nan	nan	nan	nan	nan	2.5	7.	0.2916666666...	nan
'1aNi6QhnX2hht76bUW1ESz'	nan	nan	nan	nan	nan	nan	0.2916666666...	2.5	7.	nan
'1aNi6QhnX2hht76bUW1ESg'	nan	nan	nan	nan	nan	nan	1.	5.	3.9999999999...	nan
'1aNi6QhnX2hht76bUW1ESM'	nan	nan	nan	nan	nan	nan	1.	5.	3.9999999999...	nan
'1PI6e6Y2z9LeKjNK4\$URTi'	nan	nan	nan	nan	nan	nan	1.	2.	2.	nan
'1aNi6QhnX2hht76bUW1EQm'	nan	nan	nan	nan	nan	nan	1.	5.	3.9999999999...	nan
'1aNi6QhnX2hht76bUW1ESd'	nan	nan	nan	nan	nan	nan	1.	3.0019685039...	7.	nan
'1aNi6QhnX2hht76bUW1ESQ'	nan	nan	nan	nan	nan	nan	0.2916666666...	3.0019685039...	7.	nan
'1aNi6QhnX2hht76bUW1ESP'	nan	nan	nan	nan	nan	nan	0.2916666666...	3.0019685039...	7.	nan

