A knowledge model-based BIM framework for automatic code-compliant quantity take-off

Article *in* Automation in Construction · January 2022

DOI: 10.1016/j.autcon.2021.104024

4 authors:

Hao Liu
The Hong Kong University of Science and Technology

Jack C. P. Cheng
The Hong Kong University of Science and Technology

Vincent J.L. Gan
National University of Singapore

Shanjing Alexander Zhou
Imperial College London

[Title Page]

A Knowledge Model-based BIM Framework for Automatic Code-compliant Quantity Take-off

Hao Liu[a], Jack C.P. Cheng[a,*], Vincent J.L. Gan[b,*], Shanjing Zhou[c,d]

[a] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, 999077, Hong Kong.

[b] Department of the Built Environment, National University of Singapore, 117566, Singapore.

[c] MES Group, 999077, Hong Kong.

[d] Centre for Systems Engineering and Innovation, Department of Civil and Environmental Engineering, Imperial College London, London, SW7 2AZ, UK.

*Corresponding authors.

Email: cejcheng@ust.hk (Jack Cheng); vincent.gan@nus.edu.sg (Vincent Gan)

ABSTRACT

The results of quantity take-off (QTO) based on building information modeling (BIM) technology rely heavily on the geometry and semantics of 3D objects that may vary among BIM model creation methods. Furthermore, conventional BIM models do not contain all the required information for automatic QTO and the results do not follow the descriptive rules in the standard method of measurement (SMM). This paper presents a new knowledge model-based framework that incorporates the semantic information and SMM rules in BIM for automatic code-compliant QTO. It begins with domain knowledge modeling, taking into consideration QTO-related information, semantic QTO entities and relationships, and SMM logic formulation. Subsequently, linguistic-based approaches are developed to automatically audit the BIM model integrity for QTO purposes, with QTO algorithms developed and used in a case study for demonstration. The results indicate that the proposed new framework automatically identifies the semantic errors in BIM models and obtains code-compliant quantities.

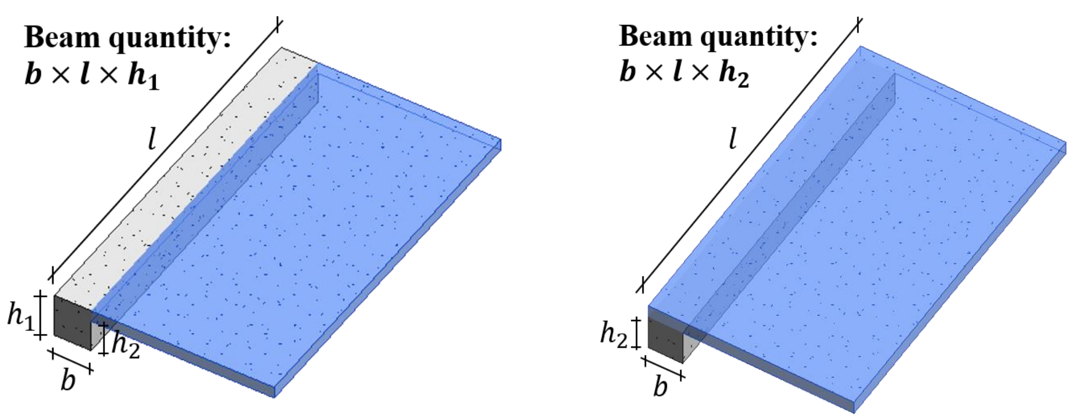
Keywords:

Automatic semantic auditing, Building information modeling, Code-compliant, Data model, Knowledge model-based framework, Quantity take-off, Semantic representation

1. INTRODUCTION

Quantity take-off (QTO) is one of the most fundamental activities in a construction project [1,2]. The material quantities have substantial impacts on the outcome of the cost estimation [3]. Quantity surveyors use their experience to interpret 2D design drawings and manually or semi-automatically calculate the material quantities according to the descriptive rules in the standard method of measurement (SMM), which is a time-consuming and error-prone process [4–6]. The building information modeling (BIM) technology has had revolutionary impacts on the conventional QTO process as quantities can be taken automatically from 3D design models [5]. The BIM-based method makes the QTO process more automated and reliable than the conventional methods using 2D drawings [5,7,8].

In addition to QTO automation, the accuracy of the output material quantities from BIM is another major concern [9–11]. The accuracy of QTO can be attributable to two fundamental aspects. Firstly, the limitations of BIM data hinder a smooth QTO. In the BIM domain, open standard Industry Foundation Classes (IFC) can express the geometric and semantic information of building elements. However, the IFC-based BIM model does not incorporate all the necessary information for conducting accurate QTO. It lacks fine-grained definitions for QTO-specific purposes. For example, the coffered and troughed slab specified in the SMM [12] is missing in *IfcSlabTypeEnum*. On the other hand, the information stored in BIM models may not be sufficient for QTO [13]. The lack of QTO-related information impedes the performing of measurement rules. For example, missing concrete grade information may result in inaccurate concrete quantities because the SMM specifies different measurement methods when elements at joints have the same and different concrete grades [12]. Secondly, the material quantities computed from existing BIM authoring software rely heavily on the geometry of 3D objects. Such a model-based approach depends on the geometric representation of building elements without taking into consideration the calculation logic in SMM, therefore the QTO results may not be compliant with the measurement rules [14,15]. It is not uncommon to see that the output quantity for the same building element varies due to different ways of creating the design BIM model [3]. Fig 1 (a) and (b) demonstrate two different BIM model creation methods for a beam-suspended slab joint, and the impact on quantity measurement.



(a) The slab is created to the side of the beam (b) The slab is created through the beam

Fig 1. Different BIM model creation methods for a beam-suspended slab joint

As Fig 1. (a) shows, the slab is created to the side of the beam, therefore, the quantity of the beam taken from the BIM model is $b \times l \times h_1$. However, this may not be compliant with the SMM rules adopted by many commonwealth countries. As shown in Fig 2, according to SMM rules, if the concrete grades of the beam and slab are different, the beam quantity should be measured through the slab ($b \times l \times h_1$), which is the same as the geometric representation of the BIM model. Otherwise, the beam should be measured only to the soffit of the slab ($b \times l \times h_2$). The lack of the required semantic information and the formulation of calculation logic in BIM cause inaccuracies in automatic QTO and practical cost estimation.

CLASSIFICATION TABLE					MEASUREMENT RULES
18. Suspended slabs **19. Coffered and troughed slabs**			m^3	1. Horizontal 2. Sloping ≤ 15° 3. Sloping > 15°	M.12 The measurement of suspended slabs is taken across columns and beams, except where the columns or beams are of a different mix.

Fig 2. Part of the SMM descriptions for measuring slab quantities [12]

To make the results compliant with the SMM rules, extensive efforts are required. Quantity surveyors need to adjust the results exported from BIM software or engineers need to follow specific model creation methods for QS. Either approach involves considerable manual adjustments and a good understanding of QS rules, which is almost impossible in practice. Alternatively, quantity surveyors can use commercial QTO software (e.g., Glodon Cubicost [16]), where IFC (Industry Foundation Classes)-based BIM models are imported to generate quantities. However, users need to setup or adjust the rule templates in the software to have accurate results, which indeed requires users to be familiar with the settings and put considerable effort in making the adjustments. Moreover, the commercial QTO software obtains quantities in a model-based manner, which means the calculation is purely based on the geometric modeling, making the results vulnerable to BIM model creation methods. In addition to the code-compliance problem, the representation of measurement standards is another matter of concern. These are usually certain descriptive rules (e.g., "*The measurement of suspended slabs is taken across columns and beams, except where the columns or beams are of a different mix.*") [12] that are understood by domain experts but not clear to others for the design of computerized programs in facilitating the QTO process.

Therefore, the objective of this study is to develop a new knowledge model-based framework to incorporate all the required information and formulate the SMM rules in BIM for semantic richness assurance and automatic code-compliant (i.e., the quantities are compliant with the SMM) QTO. Automatic and code-compliant QTO can be undertaken for modeled and unmodeled elements regardless of the BIM model creation method. SMM is the measurement standard in commonwealth countries. Thus, the identification and formulation of SMM requirements and rules are applicable to a wide range of areas. The proposed framework starts with domain knowledge modeling via discussions with domain experts, identification of QTO-related information, establishment of a QTO semantic data model (with related entities and their relationships), and formulation of SMM calculation logic for QTO automation. Following this, an automatic semantic auditing approach based on linguistic techniques is proposed to check the integrity of design BIM models, including the data completeness and correctness for QTO applications. Based on the knowledge model as well as the complete and correct BIM data, new calculation concepts and methods are designed to integrate both geometric and semantic information for obtaining quantities of modeled and unmodeled elements comprehensively. The performance of the proposed auditing and QTO methods are verified using BIM authoring software, with illustrative examples in different QTO scenarios.

The rest of the paper is organized as follows: Section 2 introduces previous studies on automatic and code-compliant QTO. Section 3 presents the methodology of this research, including knowledge modeling, semantic auditing, and automation in building quantity measurement, followed by the computational algorithms for automatic measurement of building quantities in Section 4. Section 5 uses illustrative examples to verify the performance of the proposed framework. Section 6 concludes the whole paper with recommendations for future work.

2. LITERATURE REVIEW

To date, various studies have explored the ways to tackle the aforementioned problems. To solve the time-consuming and error-prone problems of traditional methods that are based on 2D drawings and quantity surveyors' experience [4–6], early research linked the 2D CAD drawings with the bill of quantities automatically [17]. BIM provides the ability to extract quantities from 3D design models directly and therefore is faster and more reliable [5,7,8]. Bečvarovská and Matějka [18] concluded that time is saved and errors are reduced after applying BIM, based on a case study comparing BIM-based QTO and traditional methods. By using BIM models, quantities can be automatically generated and linked with cost information to enable a quick and flexible cost estimation process [19,20]. Leveraging BIM and ontology techniques, Lee et al. [21] automatically obtained quantities and associated them with work items to generate bills of quantities. Taking advantages of BIM and web-based techniques, bills of quantities can be prepared automatically and intuitively [22].

However, despite the high demand in cost and time required in modeling sufficient details for BIM-based QTO and cost estimation [23,24], coupled with the attempts to improve productivity in the modeling process [9,25], the accuracy problem due to the modeling methods was found to be one of the main obstacles in the BIM-based QTO process [3]. Through two BIM-based QTO case studies, Firat et al. [26] suggested adopting agreed modeling guidelines to create BIM models for QTO. Monteiro and Martin [3] explored the model behavior under the constraints of QTO specifications and proposed detailed modeling guidelines to allow quantity surveyors to extract the quantities in accordance with the specifications. Zima [27] reported that the modeling methods have significant impacts on the accuracy of the quantities, and the elements should be modeled to reflect their actual construction procedures (e.g., the walls should be modeled by separate layers) to have accurate results. By comparing the results from four interior cases, Kim et al. [10] analyzed the quantity discrepancies, identified the impacting factors of such discrepancies and proposed modeling and measuring strategies to reduce them.

Another focus concerns better inference of quantities using BIM model information. Aram et al. [1] proposed a knowledge-based framework to calculate the quantities of precast concrete, with a domain knowledge base to infer and provide information for QTO and cost estimation tasks. Cho and Chun [28] used BIM object information to compute the quantities of concrete and formwork, and used Decision Tree Model and Case-based Reasoning to predict rebar quantities. Rajabi et al. [29] used a set of QS logic to estimate the quantities of the Mechanical, Electrical and plumbing (MEP) trades. Khosakitchalert et al. [2] used the information from the concept of BIM-based clash detection to subtract excessive quantities and add missing quantities to get accurate quantities for compound elements. Some scholars attempted to infer quantities of unmodeled elements using available BIM object information. For example, Lim et al [30], Khosakitchalert et al. [31], and Cepni and Akcamete [32] calculated formwork quantities mainly through subtracting intersection regions to obtain exposed surfaces. Romanovskyi et al. [33], Hyun et al. [34] and Lee et al. [35] utilized information on the concrete elements and design formulas to generate formwork designs and quantities. Similarly, Wei et al. [36] computed auxiliary materials in housing construction based on elements' information and calculation formulas. Making use of the provided BIM model information and pre-defined rules, Kim and Teizer [37] produced scaffolding designs automatically.

Further, research has begun to emphasize the problems of inadequate information in BIM models [38–40] and conflicts with measurement rules [14,15]. Choi et al. [13] proposed an open BIM-based approach to check BIM model information availability and then calculate the quantities of structural framing. To solve the interoperability problem of BIM models from different tools, Akanbi et al. [41] proposed a bottom-top method to trace the information in the IFC files for the design of robust QTO algorithms. Khosakitchalert et al. [42] took advantage of the drywall information in the BIM model to calculate the quantities of wall framings [43]. Liu et al. developed an ontology-based system to infer non-expressed information in BIM models to support QTO for wall framings. Ma et al. [15] designed a semi-automatic system based on measurement standards in China for BIM-based cost estimation. The system incorporates the measurement rules in Chinese specifications, such as the void omission and volume reduction at intersecting elements, for the computation of building quantities. In another study by Ma et al. [44], an ontology for cost estimation specifications in China was established to classify the items in the bill of quantities. Similarly, Xu et al. [45] and Abanda et al. [46] developed and used ontology to automatically calculate quantities in accordance with the measurement standards from BIM models.

In summary, the BIM-based method can make the QTO process automatic and thus can bring considerable benefits to QS professionals in terms of time and error. However, the accuracy (i.e., compliance with measurement rules) of the BIM-based method is still a matter of concern. Various studies have tried to achieve automatic and accurate BIM-based QTO, but the incomplete information and the non-compliance with measurement rules are the main obstacles. While a few studies have explored QTO information interpretability and the code-compliance of the results, a clear, systematic, and general representation of the descriptive measurement requirements and rules is still lacking, and comprehensive rule-compliance of the results covering both modeled and unmodeled elements has not been addressed. Furthermore, even though some previous studies and software tools in the market attempted to incorporate SMM rules into the automatic QTO, their approaches are still limited to the model-based method, which means the computation of material quantities is subject to appropriate geometric representations of design BIM models. No research has focused on the robustness to model creation methods for the same structure and potential unintended information mistakes in the QTO process, which would also cause inaccuracies in the results. Therefore, with the development of a framework that includes a knowledge model incorporating SMM requirements and rules, semantic auditing approaches, and new QTO concepts and algorithms, this study ensures BIM model semantic richness for QTO purposes and generates code-compliant quantities that are robust to different BIM model creation methods. It contributes to the following:

- Enhancing the interoperability and openness of QTO-related information through the establishment of a semantic data model and the formulation of measurement rules.
- The principles of the proposed semantic auditing approach based on linguistic techniques can be applied not only for QTO but also for other purposes to ensure semantic richness in the applications.
- The newly designed QTO concepts and algorithms can provide quantity surveyors with automation, accuracy, and robustness in the QTO process for both modeled and unmodeled elements.

3. METHODOLOGY

As shown in Fig 3, the proposed methodology framework starts with knowledge modeling by leveraging SMM rules and domain knowledge from quantity surveyors in order to identify the information requirements for automatic QTO. Following this, a semantic data model is established to represent the QTO-related entities, attributes and their relationships. In addition, a rule library is constructed to represent the calculation logic from SMM for supporting the automatic QTO. Furthermore, an automatic semantic auditing method based on linguistic techniques is proposed to check the information completeness and correctness in the design BIM models. Last but not least, the geometric and semantic information from the audited BIM model is utilized as the input for automatic measurement. It first goes through the category discrimination logic, such as the classification between columns and walls based on the aspect ratio and the consideration regarding bottom and top formworks to horizontal elements based on the sloping. For modeled elements, the semantic information (e.g., concrete grade) is utilized to determine the calculation scope (e.g., other elements are measured to the soffit of the slab or through the slab). Based on such judgments, the geometric information (e.g., element solid, cross-section) is extracted as the input for the corresponding measurement modules where new calculation concepts and methods are designed to perform the code-compliant QTO. The extracted geometric information of the modeled elements is also utilized to support the logical deduction in the calculation for the quantities of unmodeled elements such as formwork. Based on logical deduction, the quantities in terms of dimensions and/or numbers can be obtained. Finally, after processing the geometric and semantic information in the BIM model comprehensively, the measurement modules output the quantities of both the modeled and unmodeled elements in compliance with the measurement rules. In this research, the Hong Kong Standard Method of Measurement 4 (HKSMM4) [12] is selected as an illustrative example. The major rules and ideas are similar in the commonwealth countries (the UK, Singapore, etc.) that adopt SMM and/or CEMS as the measurement standards. Details of the methodology are provided in the following subsections.

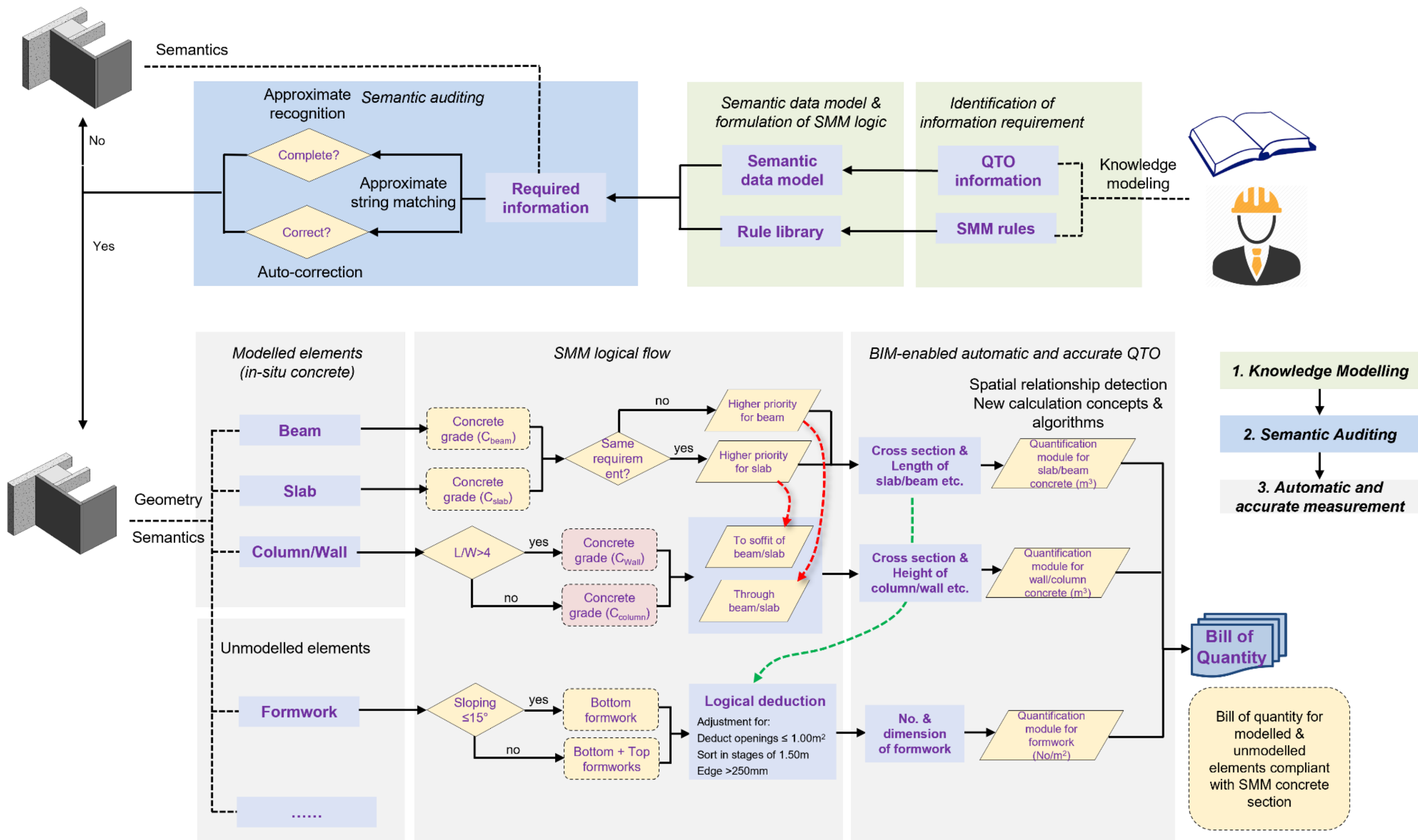


Fig 3. Overview of the proposed methodology framework

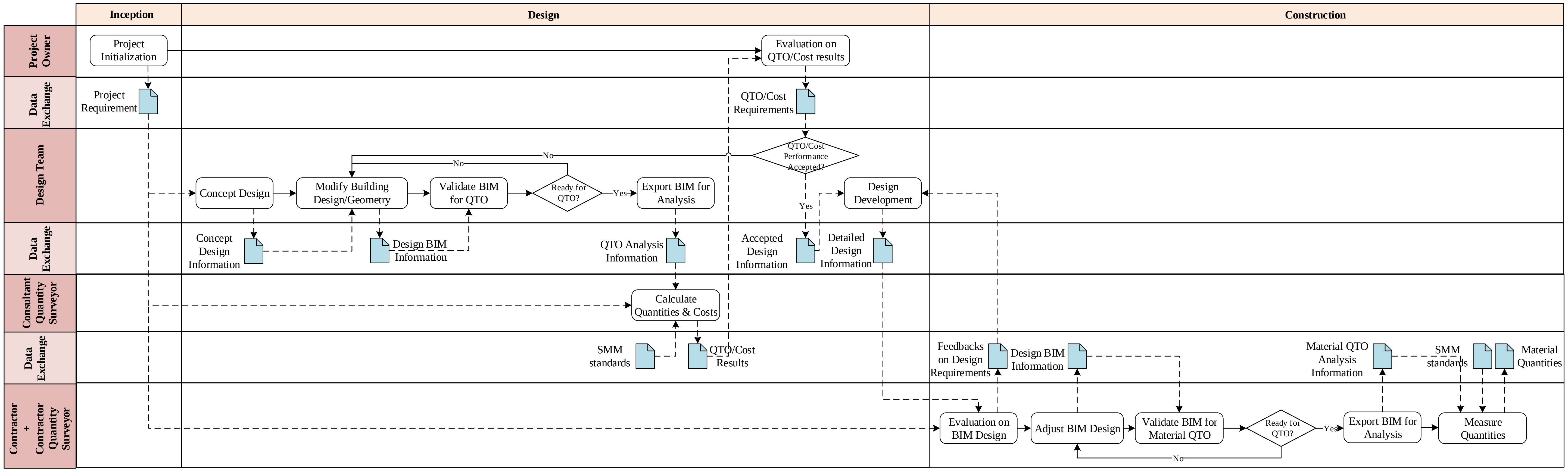


Fig 4. Typical QTO process map showing the information flow and data exchange among different tasks and stakeholders

3.1. Knowledge Modeling

3.1.1. Identification of information requirements

To identify the required information for QTO, Fig 4 shows a typical QTO process map of a construction project that includes the QTO information flow and data exchange among different tasks and stakeholders. It involves three main stages (i.e., inception, design, and construction, from left to right columns) and four major groups of stakeholders (i.e., project owner, design team, consultant quantity surveyor who provides QTO and cost estimation for the project owner, contractor and contractor quantity surveyor, from top to bottom rows). At the inception stage, the project owner initializes the project and passes the project requirements such as the project scope and QTO/cost target to the design team for the concept design. After the concept design is completed, the architects define a layout with all the necessary elements and their placements, as well as the space objects and configurations. The information requirements in terms of the project, site, building, building stories and 3D geometry of the basic elements should be met. Next, the concept design information is delivered for QTO preparation. The designers provide necessary supplements or modifications of the design information such as construction types, building geometry and design parameters for the QTO purpose. The prepared QTO design information is then validated to ensure that the design incorporates the QTO requirements. The design ready for QTO is sent to the consultant quantity surveyors. The QTO analysis information should meet the exchange requirements from design to QTO at this stage, which should include the geometry of elements, construction types, identity properties and any other necessary modifications to the building.

Based on the received design information, the consultant quantity surveyors take off the material quantities and estimate the costs according to the owner's requirements and SMM rules. The project owner checks whether the QTO and/or cost estimated meet the project requirements (e.g., project scope, QTO, and cost targets). The cost of the design is deemed acceptable if the project requirements are met; otherwise, the designers need to modify the design solutions until the requirements are met. If the concept design is accepted, the design information is delivered to structural and MEP engineers for design development. This stage includes the representations of different kinds of building objects in varying shapes, sizes, locations, etc., and the attached non-graphical properties. It goes through the evaluation of contractors and may be modified over several rounds until confirmation. Next, the contractor adjusts the received design information for material QTO. The corresponding design information is validated to ensure that it is ready for QTO in terms of the required information. Finally, the contractor quantity surveyors take off the material quantities and estimate the construction cost based on the detailed material information, construction method statement, SMM rules, etc.

Through this identification process, the QTO-related information, as well as its transfer among different major stakeholders and stages, is recognized, providing the basis for the representation of QTO semantics. The process map links the information flow and data exchange among different stakeholders towards material QTO, and the required information at different stages can then be identified to determine the semantic data model.

3.1.2. Semantic data model

Based on the process map, all the required information for automatic QTO in accordance with SMM can be taken to establish the data model. The data model represents all the QTO-related entities, their attributes, and semantic relationships, which are then leveraged to formulate the calculation logic in SMM. In this paper, typical modeled and unmodeled elements in HKSMM4 Section VII: Concrete Works [12] are selected to build a semantic data model in common scenarios of in-situ concrete structures for illustrative purposes, using relevant entities, attributes, and relationships in the latest official IFC standard, IFC4_ADD2_TC1 [47], as shown in Fig 5. Entities/attributes that are required for QTO but are missing in the standard are highlighted in red.

Part I shows the most fundamental modeled building elements in SMM. Element attributes and relationships are generalized in the supertype *IfcBuildingElement*. Each element should have a unique identifier and unit, which are the *Reference* and *Unit* properties in *Pset_EnvironmentalImpactIndicators*. In addition, concrete grade information (i.e., the *StrengthClass* property in *Pset_ConcreteElementGeneral*) is needed for in-situ concrete elements in order to formulate the SMM calculation logic under the situations of using the same and different concrete grades between slabs and other elements (i.e., beams, columns, and walls are measured to the soffit of the joined slabs if their concrete grades are the same as those of the slabs, otherwise, they are measured through the slabs. More details can be seen in Section 3.1.3). *IfcProductDefinitionShape* and

IfcShapeRepresentation represent that the modeled elements may have varying shapes with different dimensions, and *IfcQuantityVolume* indicates that those modeled concrete elements are measured in volume. *IfcBeam*, *IfcColumn*, *IfcWall*, *IfcSlab* are four subtypes of *IfcBuildingElement* and inherit the common attributes from it. Of note is that each of the modeled elements may have unique geometric features and/or be subject to specific SMM rules, thus they may consist of additional attributes to meet the information requirements for automatic QTO. For example, in the SMM rules, top formworks are required for horizontal elements (e.g., beam and slab) if the slopes are greater than 15°. Subsequently, the *Slope* and *PitchAngle* properties in *Pset_BeamCommon* and *Pset_SlabCommon* are needed for *IfcBeam* and *IfcSlab*, respectively. Further, for these modeled elements, the SMM rules have specific definitions that should be classified separately, such as suspended slabs and coffered and troughed slabs, which are specified through the predefined or extended types in *IfcSlabTypeEnum*, *IfcWallTypeEnum*, and *IfcBeamTypeEnum*.

Part II represents the void information in building elements. *IfcRelVoidsElement* associates *IfcBuildingElement* and *IfcOpeningElement* as well as the volume. This information is required because the SMM has special measurement rules for voids. For example, voids less than 0.5 m^3 are not subtracted from concrete quantities. Similarly, no subtractions are made in formwork quantities for openings less than 1.0 m^2.

Part III describes the QTO-related information for formwork. Formwork can be instantiated as a product through *IfcConsturctionProductResource*. It is an unmodeled element, so the calculation of quantities relies on the information from the assigned modeled elements. It is measured in different units (e.g., m^2, Number) in the SMM rules (e.g., formworks to cantilever ends are counted in numbers), which is expressed by *IfcPhysicalSimpleQuantity* such as *IfcQuantityArea* and *IfcQuantityCount*. Specifically, SMM requires different kinds of formworks (e.g., left-in and permanent) to be measured separately. Such fine-grained classifications are required through *FormworkType* properties for the distinction purpose.

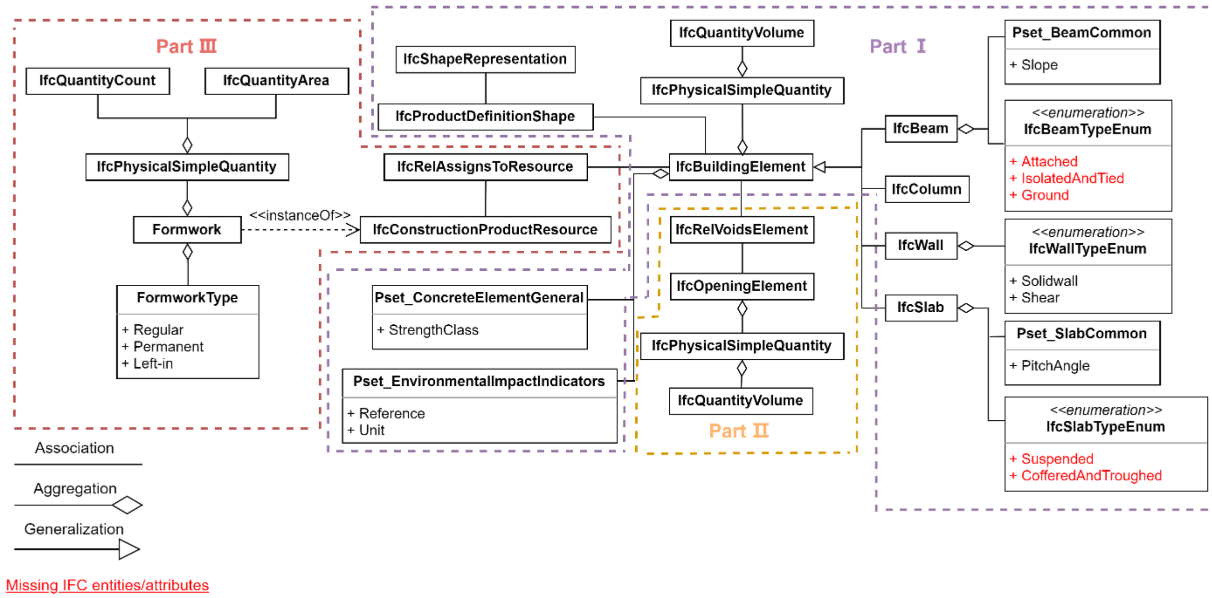
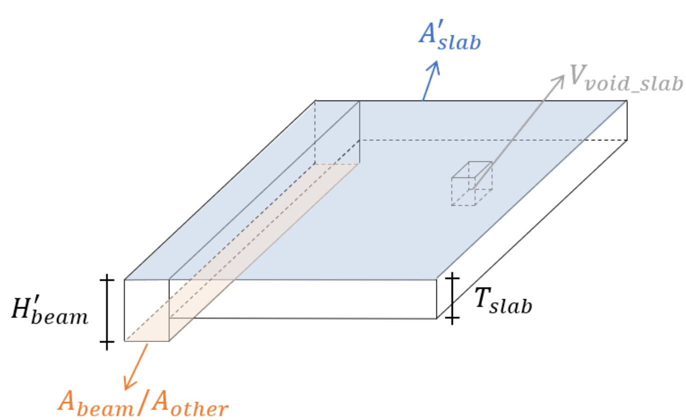


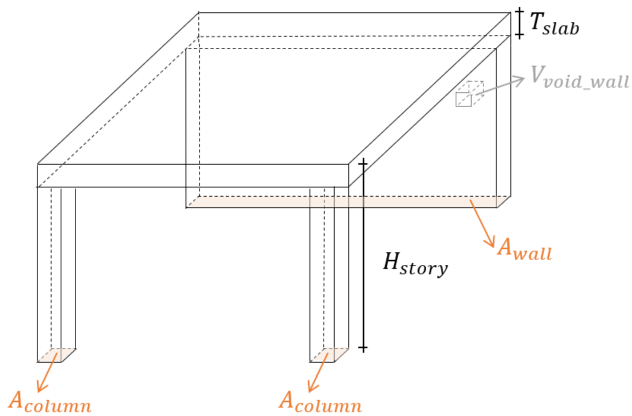
Fig 5. Semantic data model for QTO

3.1.3. Mathematical Formulation of SMM Logic

The semantic data model provides the necessary information to help formulate the calculation logic in SMM. In this paper, a rule library is developed to express the logic of the modeled elements and unmodeled elements aforementioned. Fig 6 shows the typical joints of in-situ concrete elements. Detailed formulations are as follow.



(a) Slab-beam joint



(b) Slab-column/wall joint

Fig 6. Typical joints of in-situ concrete elements

- Generic formula for slabs and beams

For QTO of in-situ concrete slabs, the key principle is to determine the priorities between the slabs and other concrete elements. For example, in a beam-suspended slab joint, if the concrete grades of the beam and slab are the same, the slab quantity should be measured through the beam. Otherwise, it should be measured only to the side of the beam. As such, the automatic QTO for slabs and beams follow the equations below:

$$V_{slab} = A_{slab} \times T_{slab} - V_{void\_slab} \tag{1}$$

$$A_{slab} = \begin{cases} A'_{slab}, & C_{slab} = C_{other} \\ A'_{slab} - A_{other}, & C_{slab} \neq C_{other} \end{cases} \tag{2}$$

$$V_{void\_slab} = \begin{cases} 0, V_{void\_slab} \leq 0.05m^3 \\ V_{void\_slab}, otherwise \end{cases} \tag{3}$$

where V_{slab} represents the volume of slab (m^3), A_{slab} is the area of slab (m^2), A'_{slab} refers to the area of slab soffit across joined elements (m^2), A_{other} is the soffit and cross-section areas of joined elements (m^2), T_{slab} is the thickness of slab (m), and V_{void_slab} stands for the void space in slab concrete (m^3). C_{slab} and C_{other} are the concrete grades of the slab and the joined elements, respectively.

$$V_{beam} = A_{beam} \times H_{beam} - V_{void} \tag{4}$$

$$H_{beam} = \begin{cases} H'_{beam} - T_{slab}, & C_{slab} = C_{beam} \\ H'_{beam}, & C_{slab} \neq C_{beam} \end{cases} \tag{5}$$

$$V_{void\_beam} = \begin{cases} 0, V_{void\_beam} \le 0.05m^3 \\ V_{void\_beam}, otherwise \end{cases} \tag{6}$$

in which V_{beam} refers to the volume of beam (m^3), A_{beam} is the area of beam soffit (m^2), H_{beam} is the height of beam (m), H'_{beam} is the height from the soffit of the beam to the top surface of the slab (m), T_{slab} refers to the thickness of joined slab (m), and V_{void_beam} stands for the void space in beam concrete (m^3). C_{slab} and C_{beam} are the concrete grades of the beam and the joined slab, respectively.

- Generic formula for columns and walls

For QTO of in-situ concrete walls and columns, the categories should be first classified before the calculation. For example, if the width of a column exceeds four times its thickness, it is classified as a wall. The main calculation principle is similar to that of beams. If the column/wall and slab are of a different mix, the volume of the column/wall should be measured through the slab (i.e., measured up to the top of the slab). Otherwise, it is measured up to the soffit of the slab. Therefore, the equations for taking off the quantities of columns and walls are as below.

$$V_{column/wall} = A_{column/wall} \times H_{column/wall} - V_{void\_column/wall} \tag{7}$$

$$H_{column/wall} = \begin{cases} H_{story} - T_{slab}, \quad C_{column/wall} = C_{slab} \\ H_{story}, C_{column/wall} \ne C_{slab} \end{cases} \tag{8}$$

$$V_{void\_column/wall} = \begin{cases} 0, V_{void\_column/wall} \le 0.05m^3 \\ V_{void\_column/wall}, otherwise \end{cases} \tag{9}$$

where $V_{column/wall}$ refers to the volume of column and wall (m^3), $A_{column/wall}$ is the cross-section area of column and wall (m^2), $H_{column/wall}$ stands for the height of column and wall (m), H_{story} is the height of story (m), T_{slab} represents the thickness of joined slab (m), and V_{void} is the void space in concrete (m^3). $C_{column/wall}$ and C_{slab} are the concrete grades of column/wall and the joined slab, respectively.

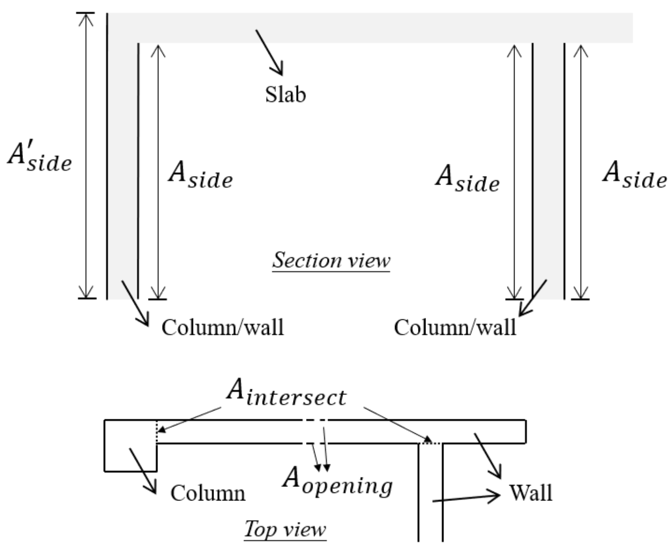
- Generic formula for formworks

For QTO of formworks, the key principles are the discrimination of formwork areas (e.g., different side formwork areas of internal and external columns), the reduction/no-reduction rules at the intersections (e.g., reduction at column-wall intersections and no-reduction at column-beam intersections), and the consideration of special positions (e.g., formwork counted in number for the cantilever ends). It should be noted that formworks are not modeled in BIM, thus their quantities should be deduced indirectly based on the geometric and semantic information of the modeled elements, taking into consideration the reduction rules. Table 1 constructs the calculation logic for formworks to the modeled elements aforementioned.

Table 1. Formulation of calculation logic for formworks to typical in-situ concrete elements

<table>
<tr><th>Unmodeled elements</th><th>Formulation for calculation of material quantities</th></tr>
<tr><td>

Formwork to column/wall



</td><td>

$$A_{fwk-c/w} = \sum A_{side} - \sum A_{opening} - \sum A_{intersect} \quad (10)$$

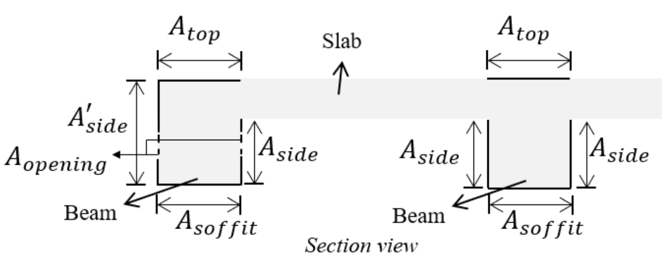
$$A_{side} = \begin{cases} A_{side\_1}, exterior\ side\ of\ external\ column/wall \\ A_{side\_0}, otherwise \end{cases} \quad (11)$$

$$A_{opening} = \begin{cases} 0, A_{opening} \leq 1.00m^2 \\ A_{opening}, otherwise \end{cases} \quad (12)$$

where $A_{fwk-c/w}$ – Area of formwork to column/wall (m^2)
A_{side} – Side areas that need formwork (m^2)
A_{side_0} – Side area measured up to the soffit of joined slab (m^2)
A_{side_1} – Side area measured up to the top of joined slab (m^2)
$A_{opening}$ – Area of opening (m^2)
$A_{intersect}$ – Area of column-wall or wall-wall intersection (m^2)

</td></tr>
<tr><td>

Formwork to beam



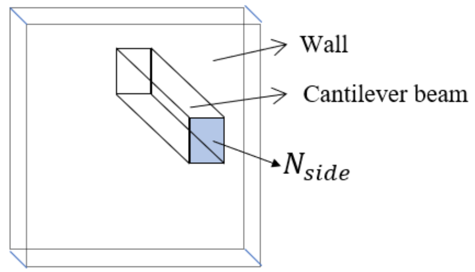
</td><td>

$$A_{fwk-beam} = \sum A_{side} + \sum A_{soffit} + \sum A_{top} - \sum A_{opening} \quad (13)$$

$$A_{side} = \begin{cases} N_{side}, cantilever\ end \\ A_{side\_1}, exterior\ side\ of\ edge\ beam \\ A_{side\_0}, otherwise \end{cases} \quad (14)$$

$$A_{top} = \begin{cases} 0, slope \leq 15° \\ A_{top}, otherwise \end{cases} \quad (15)$$

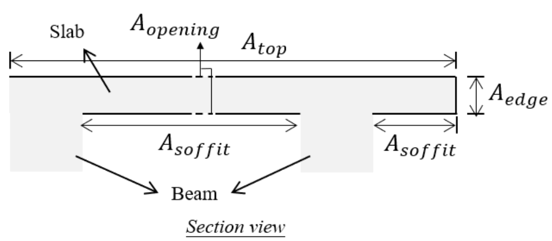
</td></tr>
</table>



$$A_{opening} = \begin{cases} 0, A_{opening} \leq 1.00m^2 \\ A_{opening}, otherwise \end{cases} \quad (16)$$

where $A_{fwk-beam}$ – Area of formwork to beam (m^2)
A_{side} –Side areas that need formwork (m^2)
N_{side} – Number of side (Nr)
A_{side_0} – Side area measured up to the soffit of joined slab (m^2)
A_{side_1} – Side area measured up to the top of joined slab (m^2)
A_{soffit} – Soffit area of beam (m^2)
A_{top} – Top area of beam (m^2)
$A_{opening}$ – Area of opening (m^2)

Formwork to suspended slab



$$A_{fwk-slab} = \sum A_{edge} + \sum A_{soffit} + \sum A_{top} - \sum A_{opening} \quad (17)$$

$$A_{top} = \begin{cases} 0, slope \leq 15° \\ A_{top}, otherwise \end{cases} \quad (18)$$

$$A_{opening} = \begin{cases} 0, A_{opening} \leq 1.00m^2 \\ A_{opening}, otherwise \end{cases} \quad (19)$$

where $A_{fwk-slab}$ – Area of formwork to slab (m^2)
A_{edge} – Edge area of slab if applicable (m^2)
A_{soffit} – Soffit area of beam
A_{top} – Top area of beam
$A_{opening}$ – Area of opening (m^2)

Based on the established knowledge model, the BIM model is audited to ensure the QTO semantic richness and is utilized to obtain automatic and code-compliant quantities. The semantic auditing and automation in quantity measurement are illustrated in the next two sections.

3.2. Linguistic-based Approach for Automatic Semantic Auditing

Following the modeling of the necessary domain knowledge, procedures are developed to perform BIM model semantic auditing to ensure that complete and correct information is provided for the subsequent QTO. As shown in Fig 7, the semantic attributes are extracted from the design BIM models. The computerized procedures can automatically identify missing information and unintended textual errors and inform users to input more data if the BIM model does not contain adequate information for QTO or to correct the information if it is expressed inappropriately.

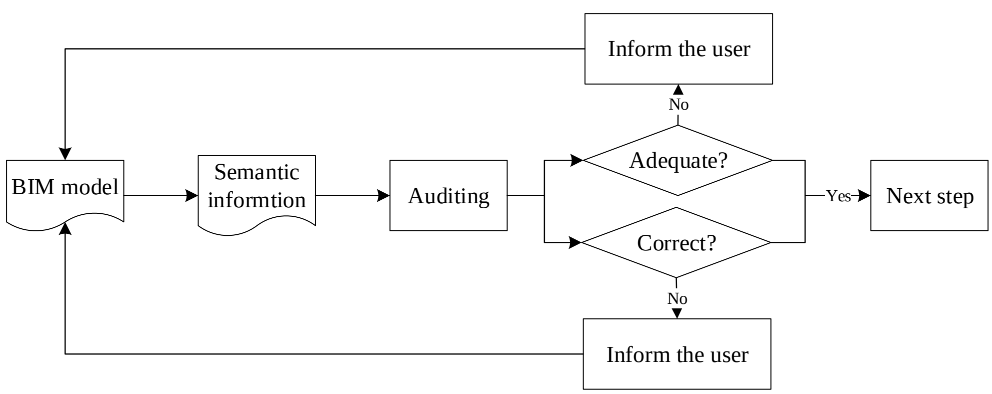


Fig 7. Overview of the proposed BIM model semantic auditing process

Fig 8 shows the developed procedure for auditing the information completeness in BIM models. The design BIM model information is first extracted. Then, null values and the associated parameters are removed before matching the parameter names with the required information derived from the established knowledge model in order to filter parameters with names existing but with values missing. The required information is concluded as existing if it is matched with the parameter name in the extracted BIM model information or the Levenshtein distance between these two compared strings is less than or equal to 1. The Levenshtein distance is a string metric measuring the similarity between two strings [48]. It indicates the minimum number of single-character edits (e.g., delete, insert, modify) when changing from one string to another. The smaller it is, the more similar are the two compared strings. This setting enables an approximate string matching so that the auditing process does not miss the model parameters that have slight differences with the defined required ones but express the same meanings. For example, the QTO process requires concrete grade information, while the *concrete grade* parameter may be misspelled as *concete grade*. In such a case, this parameter can still be recognized as existing instead of missing, which reflects the fact of the model.

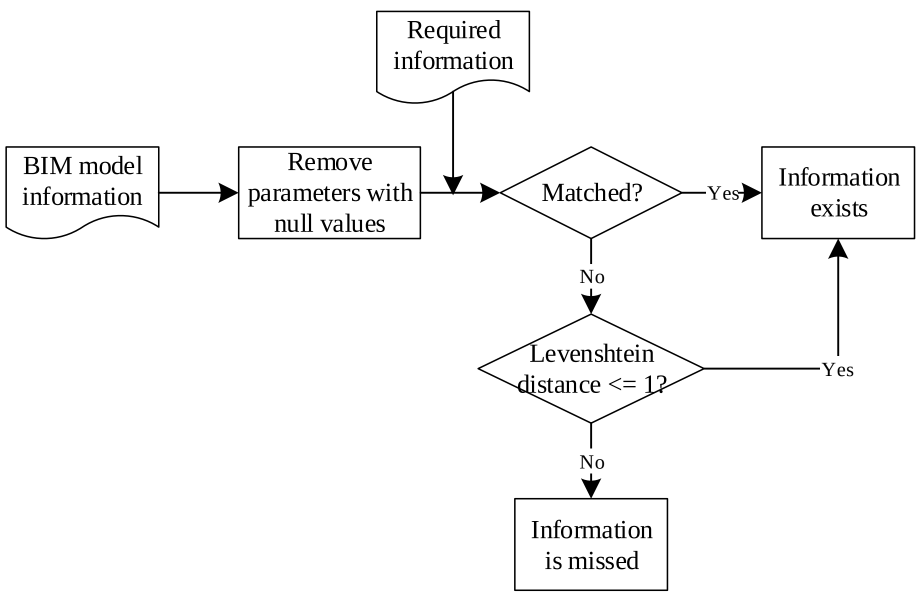


Fig 8. Auditing information completeness

Details of the information correctness auditing are shown in Fig 9 with an example RC_Sllab_200mm. First, the textural parameters are separated into multiple words by a separator list that contains common punctuation such as underline, dash, and space. The separators from the input are kept in a separator container for re-joining the words later. A linguistic checker equipped with different customized domain dictionaries is then used to check the correctness of each word. If the result is false, candidate corrections will be generated if the Levenshtein distances between the word and the suggested corrections from the dictionaries are less than or equal to 1. This setting is designed to filter numerical parts (e.g., 200mm) that are assumed to be correct since the spelling errors of such parts are less likely to happen, and checking this part may introduce unnecessary complexity. Subsequently, if there are multiple options for correcting a misspelled parameter, the category / type of the parameter in the BIM model is obtained. Then, the corrected form is selected from the corresponding dictionary. For example, the misspelling, *bf.*, in the type parameter has two correction options, *bg.* (i.e., QS abbreviation for *bearing*) and *bk.* (i.e., abbreviation for *brick*). *bg.* is selected as the corrected form instead of *bk.*, because the misspelling is type information, and *bg.* is in the type dictionary, while *bk.* is in the material dictionary. The corrected forms are added to the word container together with the original forms in the previous steps. Finally, the words kept in the word container are joined with the separators kept earlier to generate the corrected results. Note that single words and correct values also work in this procedure.

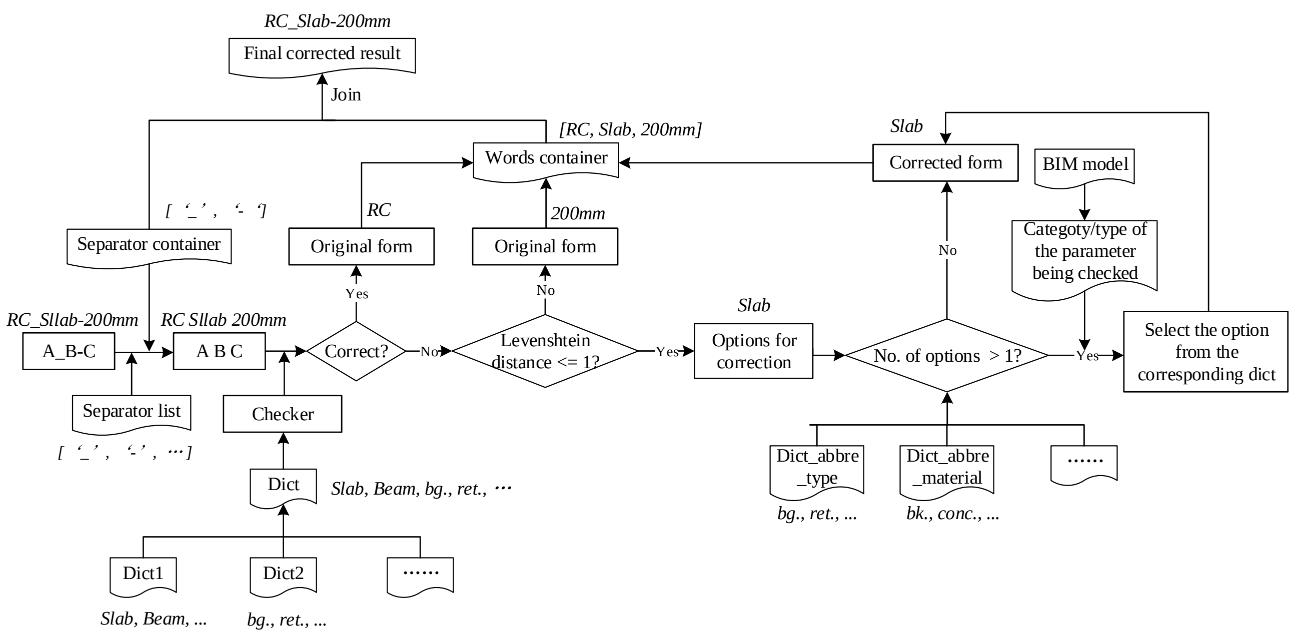


Fig 9. Auditing information correctness

3.3. Automatic and Accurate Building Quantity Measurement

Given a semantically rich QTO model, the elements go through the SMM logic flows and the corresponding calculation modules. As shown in Fig 10, the design information (e.g., geometry, spatial relation, and material properties) is extracted from BIM models and applied with the established knowledge model to determine the required variables for the automatic QTO, rather than taking the geometric data into the calculation directly. To achieve this, new computation concepts and methods are proposed.

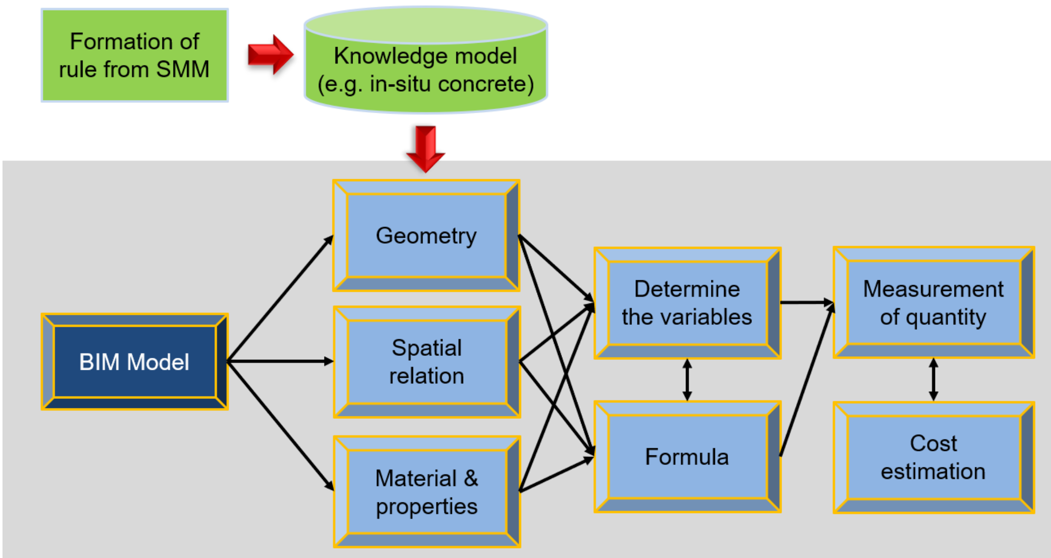


Fig 10. The process of automation in building quantity measurement

In HKSMM4 [12], quantities of the building elements depend on the information from the contact elements (e.g., the calculation priorities between slabs and other elements under the same or different concrete grades, reduction or no-reduction rules for the formwork to columns in the intersections of column-wall and column-beam). Therefore, the concept of contact detection is introduced, and the corresponding procedure is developed, as shown in Fig 11. The vertical surfaces of the target element (i.e., the element from which the quantities are taken off) are extracted and then scaled up along the z-axis with small distances to detect potential elements on it. The surfaces are then thickened on both sides with small thicknesses into corresponding solids that are used to check the intersection with other elements. Consequently, the elements in contact with the target one in the upper and surrounding directions are obtained. The information from the contact elements can be used to determine the situations mentioned above.

Instead of purely taking the geometric information (i.e., dimensions) from the BIM models for measurement, which may vary in different modeling situations and is thus unreliable (e.g., the examples in Section 1), the concept of the cutting plane is introduced.

Fig 12 shows the concept with a typical beam-suspended slab joint. The geometries are unionized as a whole. Then, based on the semantic information (e.g., concrete grade) and the spatial relationship between them, either the soffit plane of the suspended slab or the inner vertical side of the beam is utilized to cut the unionized solid to obtain the quantities of these two elements. With the help of this concept, the knowledge is embedded to some extent and the differences between different model creation methods are eliminated.

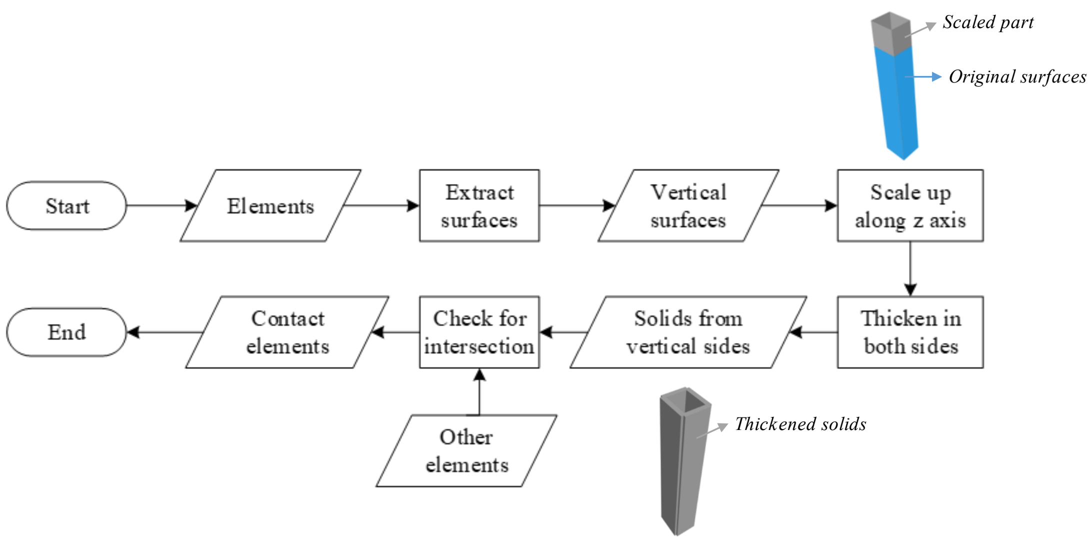


Fig 11. Contact detection

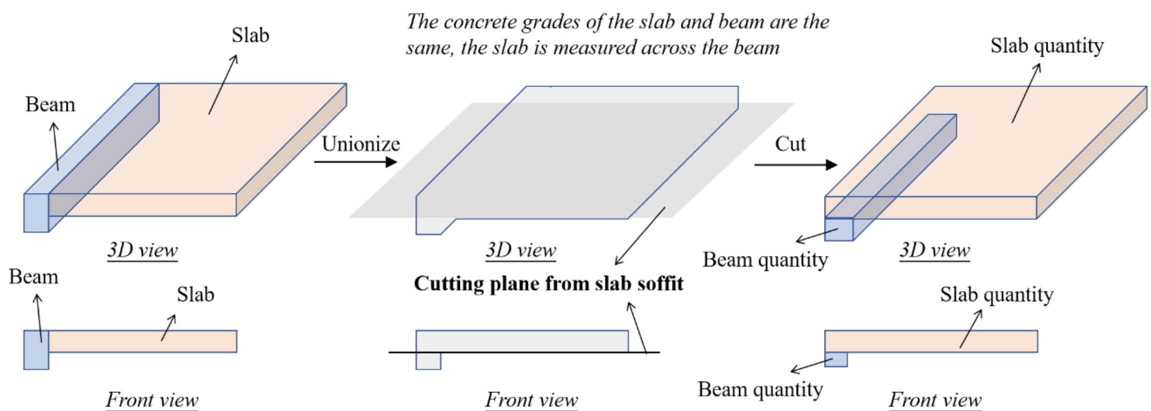


Fig 12. Concept of cutting plane

Based on these two concepts and the established knowledge model, new QTO algorithms are designed to utilize both geometric and semantic information for the automatic and code-compliant QTO, including the modeled and unmodeled elements with the concrete and formwork as examples, respectively. Eq (20) shows the general calculation equation for the concrete quantities, where V_0 is the baseline quantity that is easiest to be obtained using the introduced concepts, and Δ_i is the adjustment from other types of elements. For example, when performing the QTO on a slab, the equation should be $V_{slab} = V_{max} + \Delta_{beam} + \Delta_{column} + \Delta_{wall}$. The baseline quantity V_0 is the maximum volume V_{max} that is measured across all the other contact elements (i.e., unionizing them together as a whole and using the soffit plane of the slab to cut it). In addition, there are three adjustments from the beam (Δ_{beam}), column (Δ_{column}), and wall (Δ_{wall}), respectively.

$$V = V_0 + \sum_i \Delta_i \tag{20}$$

Fig 13 shows the general QTO algorithm for the quantities of typical in-situ concrete elements. For the target element set, the method obtains the contact elements first and then unionizes them together as a whole. In order to get the baseline quantity, a set of cutting planes are generated to cut the unionized solid. Afterward, the geometric and semantic information would be utilized to calculate the adjustments from other types of elements with the help of the cutting planes from the target elements and the contacted elements. Finally, the output is the sum of the baseline quantity and the adjustments.

Concrete QTO Algorithm

Input: Element set to be calculated $\{E_0\}$

Other element sets $\{E_1\},\{E_2\},\{E_3\}$ // $\{E_i\} \in [\{slab\},\{beam\},\{column\},\{wall\}], i = 0,1,2,3$

for each $E_0 \in \{E_0\}$ **do**

$\{E_1'\} \leftarrow \{E_1\} \cap E_0$

$\{E_2'\} \leftarrow \{E_2\} \cap E_0$ — *Get elements in contact with* E_0

$\{E_3'\} \leftarrow \{E_3\} \cap E_0$

$E_{0i} \leftarrow \bigcup_{E_i' \in \{E_i'\}} E_i', i = 1,2,3$

$E_{0123} \leftarrow \bigcup_{i=1}^{3} E_{0i}$ — *Unionize* E_0 *and its contact elements as a whole*

$E_{0123} \leftarrow E_{0123} \cup E_0$

$V_0 \leftarrow$ Volume of E_{0123} cut by cutting planes from $E_0/\{E_{01}\}/\{E_{02}\}/\{E_{03}\}$ → *Get the baseline quantity that is easiest to be obtained*

for each $i \in \{1,2,3\}$ **do**

$\Delta_i \leftarrow$ Adjustment from dimensions and cutting planes from E_0 and $\{E_{0i}\}$ — *Get adjustments from other elements*

end for

$V \leftarrow V_0 + \sum \Delta_i$ → *Sum up the baseline quantity and adjustments to get final results*

end for

Output: $\sum V$

Fig 13. QTO algorithm of typical in-situ concrete elements

The calculation of formwork is shown in Eq (2), where A_{side} are the quantities of side formwork, $A_{bottom/top}$ are the quantities of bottom and top formwork, $A_{misc.}$ are the quantities of miscellaneous formwork for special positions and rules. For example, the equation for the formwork to beam should be $A_{beam} = A_{side} + A_{bottom} + A_{misc.}$. The A_{side} are the areas of sides measured up to the soffit or the top of the contact slabs, A_{bottom} are the soffit areas, and $A_{misc.}$ are the numbers of small sections of cantilever ends.

$$A = A_{side} + A_{bottom/top} + A_{misc.} \tag{21}$$

Fig 14 shows the general QTO algorithm for the formwork quantities. The target element set would have the sides, bottoms, and tops that need formwork. If it is not a slab, the method would obtain the areas of the sides by the contacted slabs. Whether or not the side is in contact with slabs determines whether its area is measured up to the soffit or the top of the slab. The bottom or top areas would be counted as the quantities of the bottom/top formwork. If it is a slab, in addition to the side areas, the method would obtain the bottom/top areas through the union and cut operations for the element and its contact elements. Finally, the formwork counted by number would be calculated as the miscellaneous quantities.

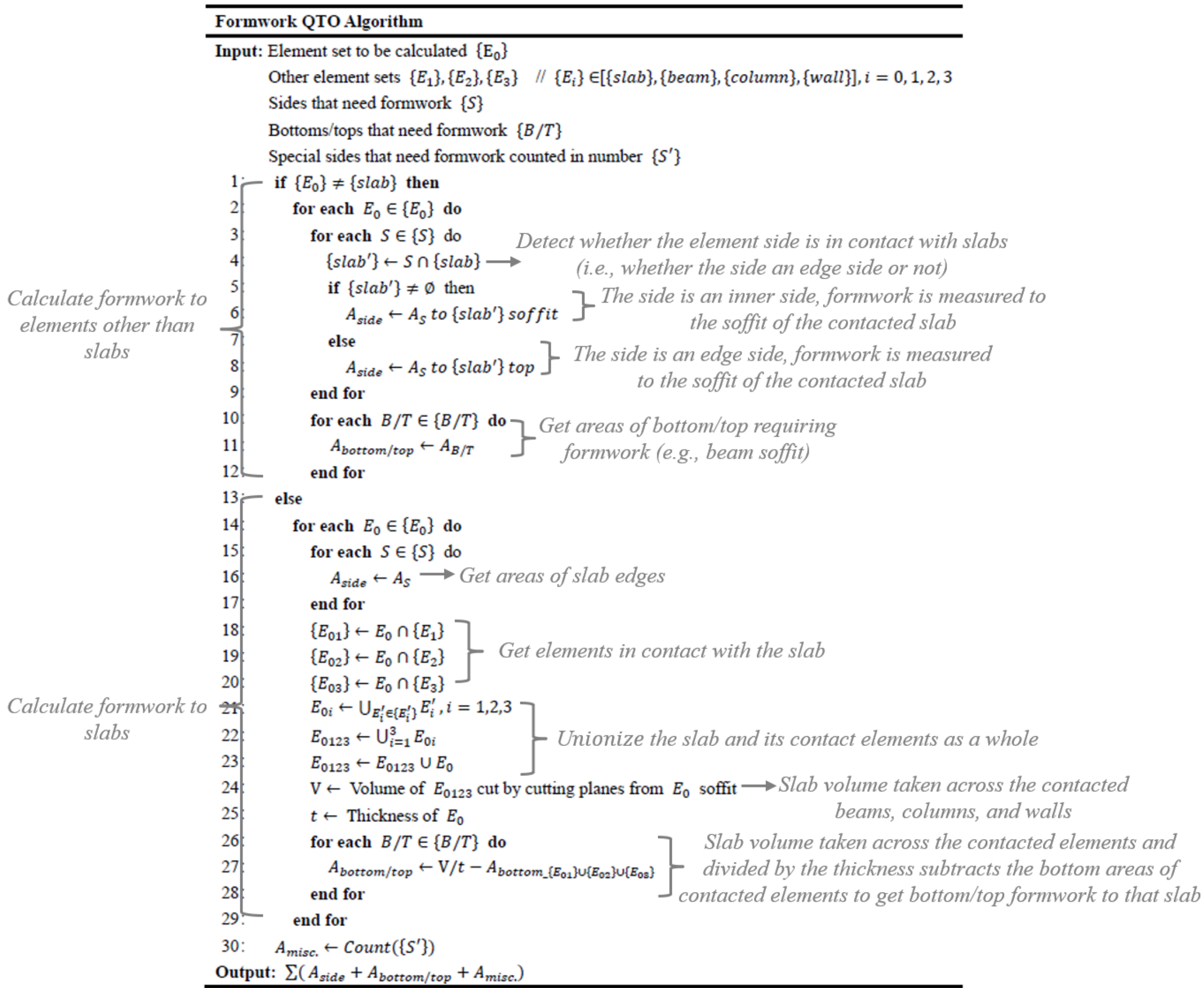


Fig 14. Formwork QTO algorithm

4. COMPUTATIONAL ALGORITHMS FOR AUTOMATIC QTO

Based on the rule library in Section 3.1.3 and the proposed concepts and algorithms in Section 3.3, detailed computational algorithms are developed to perform the actual QTO for the items. These detailed algorithms utilize both geometric and semantic information from the BIM model to determine the spatial relationships between elements and the interaction with the measurement rules, taking advantage of the designed concepts and methods. The following subsections illustrate the details, divided into the concrete and formwork parts that represent the modeled elements and unmodeled elements.

4.1. Modeled Element – Concrete

Fig 15 shows the detailed algorithm for the measurement of in-situ concrete slabs, with explanatory pictures attached for some key concepts and steps. In reference to Eqs. (1), (2), (3), the calculation equation for slabs is $V = V_0 + \Delta_{beam} + \Delta_{column} + \Delta_{wall}$, where V is the slab's quantity, V_0 is the baseline quantity, which is the maximum possible quantity measured across beams, columns and walls, Δ_{beam}, Δ_{column} and Δ_{wall} are the adjustments from these three kinds of elements, which depend on the concrete grades of the slab and the contacted elements. $A_{slab} \times T_{slab}$ in Eq. (1) is considered as the sum of $A'_{slab} \times T_{slab}$ (i.e., V_0) and the adjustments from other elements (i.e., Δ_{beam}, Δ_{column} and Δ_{wall}) which take care of A_{other} of Eq. (2). First, the algorithm detects the elements in contact with the slab and unionizes them as a whole so that models with different creation methods have the same geometric representation. Then, the soffit of the slab acts as the cutting plane to cut the integrated solid. The miscellaneous items (e.g., voids larger than 0.05 m^3, V_{void_slab} in Eqs. (1), (3)) are then subtracted and the baseline quantity is obtained. Subsequently, for each beam, the algorithm detects the contacted slabs and uses the concrete grades of the slab with the greatest thickness and the beam to determine the adjustment from the beam. If they are different, the adjustment would be $-soffit\ area\ of\ beam \times slab\ thickness$, otherwise, it is 0. For other slabs in contact with this beam, the adjustment part that is calculated by multiplying the beam's area with the slab's thickness should be deducted from their baseline quantities to avoid duplicate calculations of this part. The adjustments from columns and walls can be obtained through similar processes. Finally, the baseline quantity and the adjustment are summed to get the results.

For other elements (i.e., column, wall, beam), the computational algorithms that have similar processes are shown in appendix A.

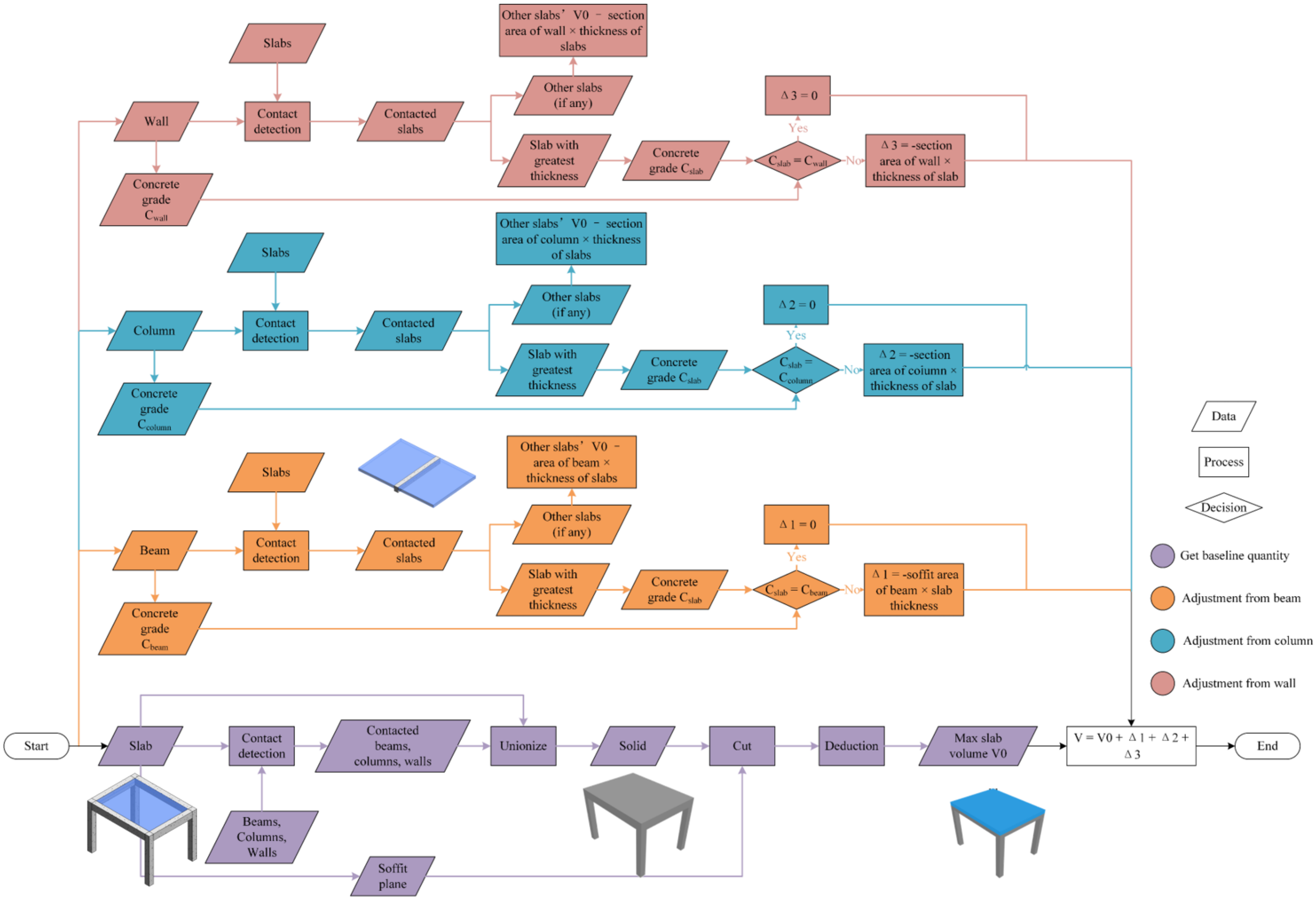
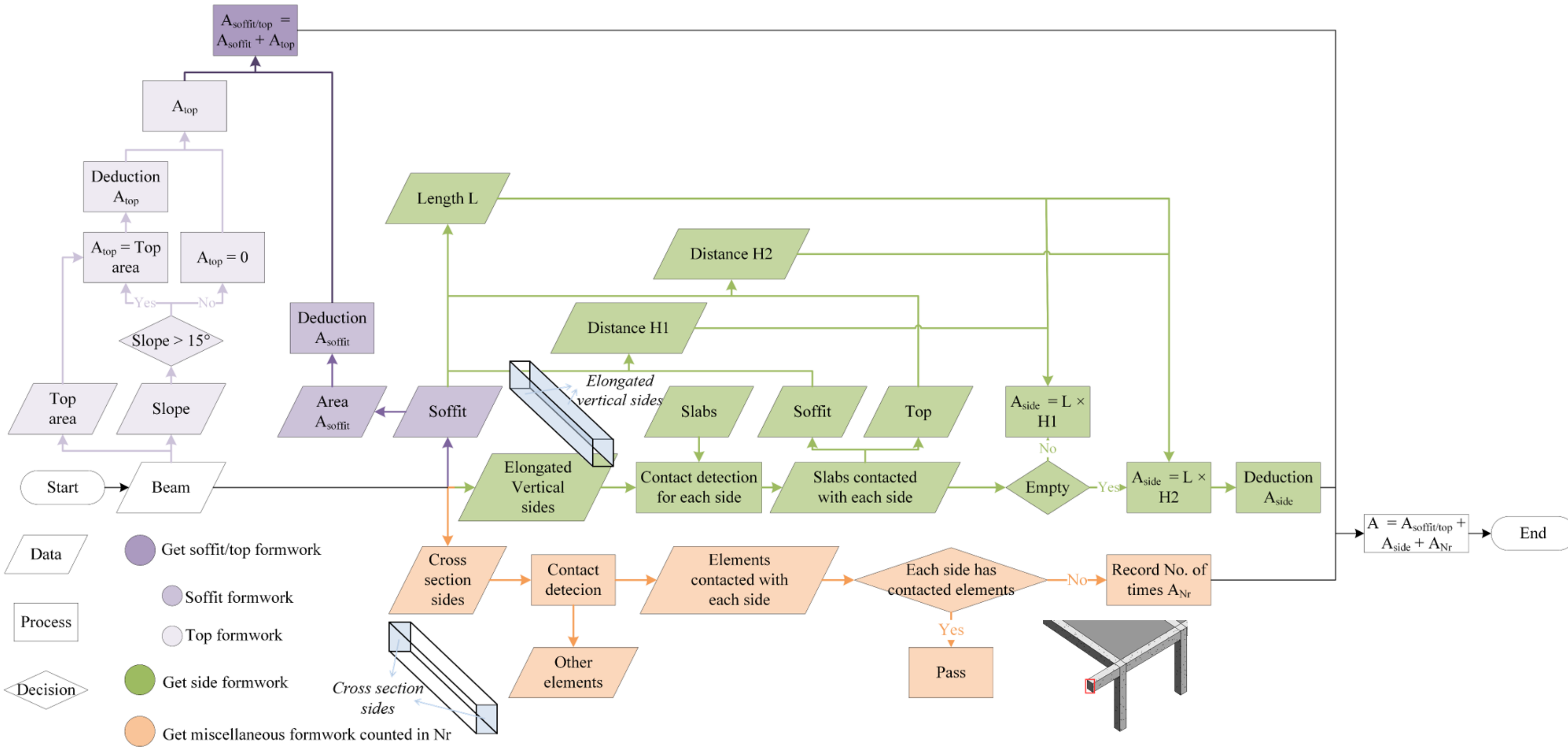


Fig 15. Algorithm for measurement of slab quantities

4.2. Unmodeled Elements – Formwork

The detailed algorithm for the measurement of formwork to beam is shown in Fig 16, and based on Eqs. (13), (14), (15), (16), the relevant equation is $A = A_{side} + A_{bottom/top} + A_{Nr}$, where A is the formwork's quantity, A_{side} is the quantity of the formwork to the beam's elongated sides, $A_{bottom/top}$ is the quantity of the formwork to the beam's bottom and top, A_{Nr} is the number of small sections of cantilever beams. In this equation, A_{side} takes care of A_{side_0}, A_{side_1} and $A_{opening}$ in Eqs. (13), (14), (16), and A_{Nr} denotes N_{side} in Eq. (14). These two items together represent A_{side} in Eq. (13). A_{soffit} and A_{top} in Eqs. (13), (15) are regarded as $A_{bottom/top}$ in this equation. First, for each beam, the algorithm extracts the elongated vertical sides and performs contact detection for each of them. No contacted slabs means that the formwork area on that side should be measured to the top of the slabs in contact with the beam. Otherwise, it is measured up to the soffit of the contacted slabs. The distances to the soffits and tops of the slabs are determined by the distances between the soffits and tops of the slabs and the soffits of beams, instead of directly taking the unreliable dimension variables (e.g., beam height) that may vary among different modeling scenarios (e.g., the examples in Fig 1). Then, the reduction items (e.g., openings larger than 1.00 m^2) are subtracted to get the side areas. For $A_{bottom/top}$, the soffit areas after subtracting the reduction items serve as A_{bottom}, while the A_{top} can be 0 or the top areas after subtracting the reduction items, depending on the sloping of the beam. $A_{bottom/top}$ is the sum of A_{bottom} and A_{top}. Further, the algorithm extracts the cross-section sides and finds the elements in contact with the sides. If there is one cross-section side without contacted elements, that beam is a cantilever beam. In that case, the formwork to such sides is counted in numbers that are obtained by recording the number of the detected cantilever ends. Finally, A_{side}, $A_{bottom/top}$ and A_{Nr} are summed to obtain the results.

For formworks to other elements (i.e., column, wall, slab), appendix B shows the computational algorithms that are similar.



615 Fig 16. Algorithm for measurement of formwork to beam

5. ILLUSTRATIVE EXAMPLES

The proposed framework is validated with three scenarios of the same model under two conditions (i.e., the slabs have the same or different concrete grades with other elements), respectively. Autodesk Revit 2021 and Dynamo 2.10 [49] are used to develop the prototype programs for illustration. The semantic auditing algorithms are implemented using Python 3.8.6., and PyEnchant 3.2.0. [50] is used as the linguistic checker to perform the exact and approximate string matchings aforementioned.

5.1. Configuration of BIM Models

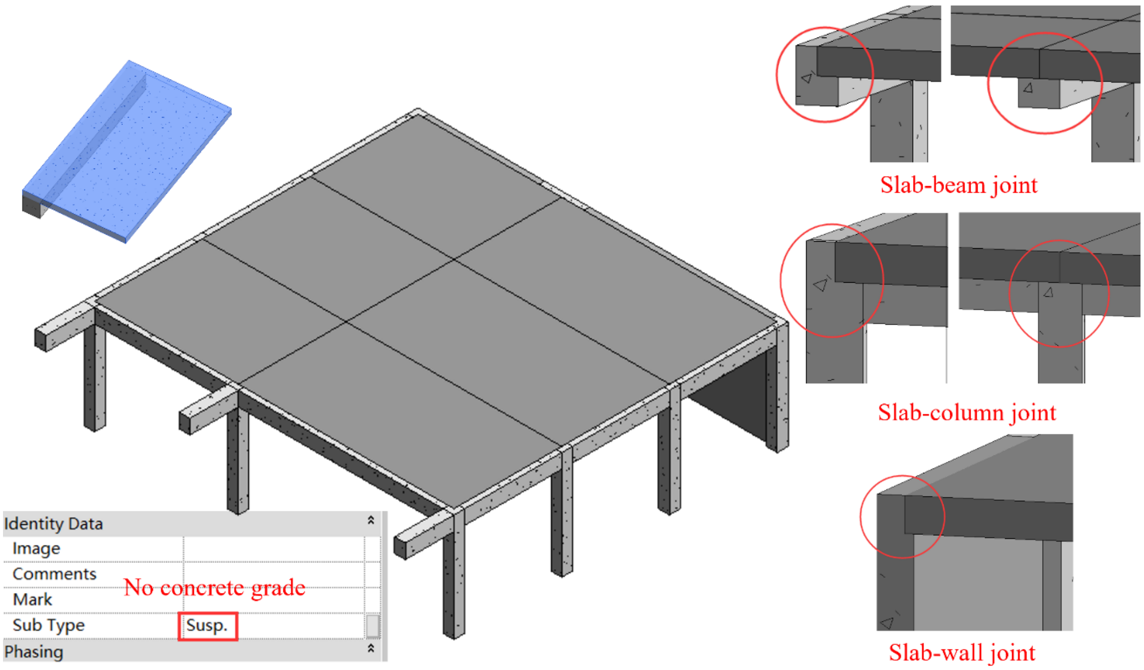
As shown in Fig 17, the model is created with reference to three different approaches in terms of the precedence between the concrete elements, as follows:

(1) The default joint where the slab takes precedence over other concrete elements, which is the default joint setting in Revit;

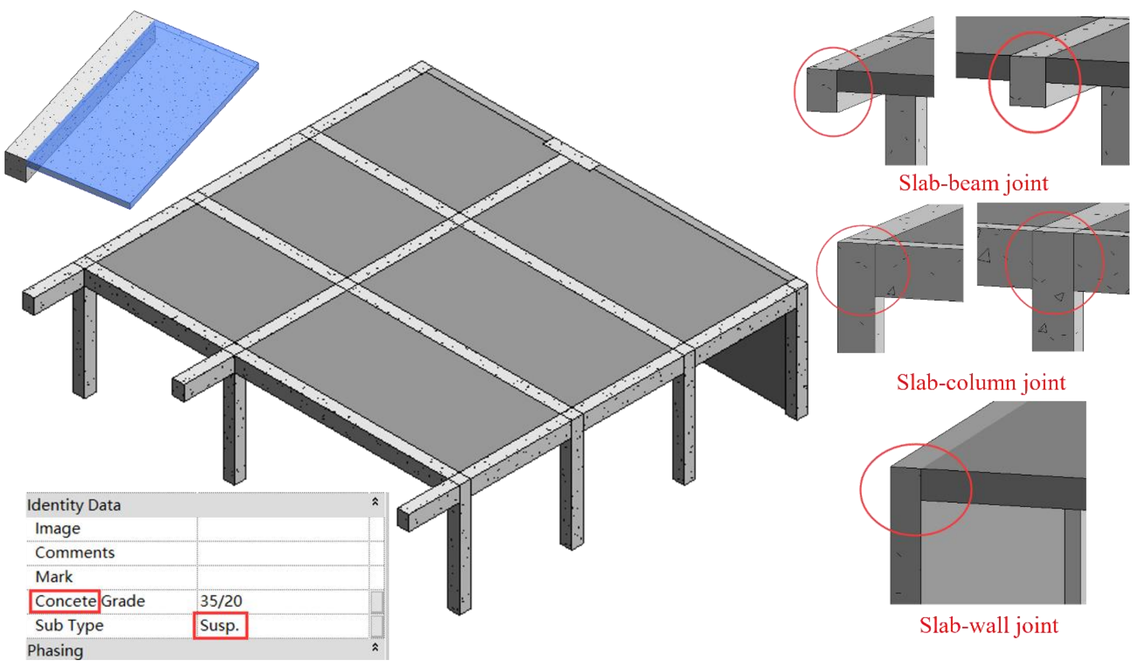
(2) The switch joint where other concrete elements take precedence over the slab;

(3) The overlap where the slab and other concrete elements are overlapped.

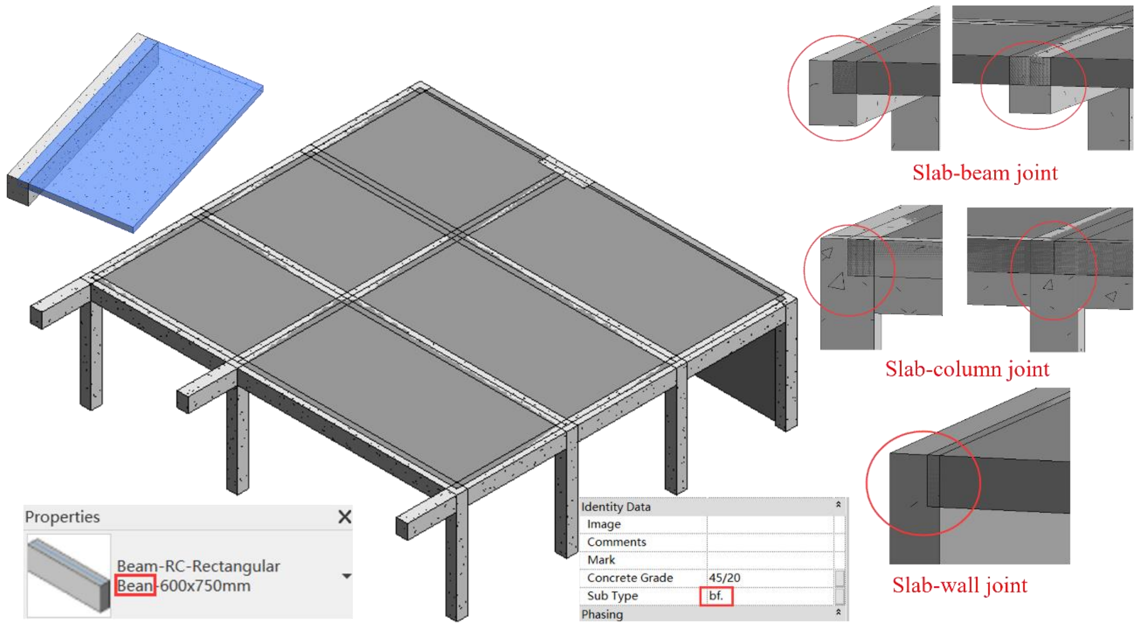
The degree of semantic information availability is intentionally designed to test the performance of the proposed semantic auditing algorithms. In the default joint model, there is no information regarding the concrete grade, and the type information is written as *susp.*, which is a common abbreviation of *suspended* adopted by quantity surveyors. In the switch joint model, while the concrete grade attribute is provided, it is misspelled as *concete*. For the overlap model, the attribute value of one type name, *beam*, is misspelled as *bean*, and the type information, *bg.*, which is the QS abbreviation of *bearing*, is misspelled as *bf.*



(a) Default joint



(b) Switch joint



(c) Overlap

Fig 17. Three modeling scenarios in Revit

5.2. Automatic Semantic Auditing

The three BIM models are checked using the proposed semantic auditing approach to ensure that complete and correct information is provided for QTO. Based on the proposed semantic data model in Fig 5, the required information for QTO is mapped with the Revit model attributes so as to generate a checklist, including the identity, geometric and semantic information. An example is shown in Table 2. Since the unit is a project-level parameter and each Revit project has its own units, there is no need to check the existence of the unit attribute for the elements. In addition to built-in parameters, users can add more project parameters or use customized families in the platform to express domain-specific information, and the naming of the same parameter varies among different modelers. For example, *b* is the beam width parameter for the system beam type, while customized beam families may use *width* to represent the beam width parameter. Similarly, *concrete grade* can be added as *conc grade* or *concrete grade*. Therefore, it is important to know how the parameters are expressed and have various kinds of domain-specific expressions for the same required information in the checklist. In the illustrative examples, the beam uses the default naming in the system. The attributes in the right column serve as the baseline to check if the information from the BIM model is complete and correct.

Table 2. Mapping required information with Revit attributes (use beam as an example)

Required information		Revit attributes
Identity	Id	Type Id
	Name	Type Name/Type
	Unit	N.A.
Geometric	Beam length	Length
	Beam width	b
	Beam height	h
	Slab Thickness	Thickness (slab)
Semantic	Concrete grade of slab	Concrete Grade/Conc. Grade (slab)
	Concrete grade of beam	Concrete Grade/Conc. Grade (beam)

As presented in Table 3, for the default joint model, the proposed algorithms satisfactorily identify the missing concrete grade information and do not misclassify the correct information (e.g., the *susp.*, which is treated as correct type information in QS, is not classified as a misspelling). Results for the switch joint model show that the information required for the QTO is complete, but the attribute name *Concrete Grade* is misspelled as *Concete Grade*. This indicates that the algorithms can still recognize the information even though the name of the attribute is misspelled unintentionally. In common keyword checking methods that perform exact matching, this typo will be reported as missed information. In addition, the proposed semantic auditing approach can automatically correct the misspelled attribute values identified in the BIM models. As the results for the overlap model shows, with part of the type name, *Bean*, being recognized as a textural error, the algorithms can also automatically correct the misspelled attribute name as *Beam*. Moreover, the misspelled type *bf.* is automatically and accurately corrected as *bg.* (i.e., QS abbreviation of *bearing*) instead of other similar forms such as *bk.* (i.e., abbreviation of *brick*). This improves the quality of the semantic information in BIM for QTO.

Table 3. Auditing results of semantic completeness and correctness in the three illustrative models

	Information completeness			Information correctness		
	ID	Adequate information?	Missed information	Parameter	Misspelling	Correction
Default joint model[1]	300310	Not	Concrete Grade/Conc. Grade	N/A	N/A	N/A
Switch joint model[1]	300310	Yes	N/A	Concrete Grade	Concete Grade	Concrete Grade
Overlap model[1]	300310	Yes	N/A	Type Name	Bean-400mm× 300mm	Beam-400mm× 300mm
	301055	Yes	N/A	Sub Type	bf.	bg.

[1]: the results for other mistake elements are the same, which are not repeated herein.

5.3. Code-compliant QTO Automation

After the design information in the BIM model is checked and corrected, prototype programs are developed to take-off the quantities of the concrete and formwork. The proposed QTO algorithms are applicable to any BIM authoring software since they are general calculation concepts. Nevertheless, to test their performance, Dynamo is utilized to develop the prototype program. Fig 18 and Fig 19 show the results of concrete and formwork from the prototype program, in comparison to the baseline (i.e., manual calculation by quantity surveyors), Autodesk Revit representing the traditional BIM-based method, and Glodon Cubicost TAS [16] representing the professional QTO software. Same-grade refers to the condition where the concrete grades of the slabs are the same as those of other elements, while different-grade is the opposite.

For the concrete results, Fig 18 shows that our proposed method outperforms the professional QTO software, and they are both better than the traditional BIM-based method. The proposed framework is stable across all of the three model creation methods regardless of having the same or different concrete grades. However, the quantities exported from the BIM authoring software and the professional QTO software vary and have relatively large deviations in most of the cases. The reason is that the BIM authoring software extracts material quantities solely based on the 3D representations of the modeled elements, which are different in the three BIM models. On the other hand, although the professional QTO software takes into consideration the SMM rules in QTO, it only handles the overlapping parts in the BIM model (i.e., intersections are not considered in reduction and no-reduction rules if the intersected elements do not overlap, as shown in Fig 20). Instead of relying on dimensions, our method utilizes different calculation concepts (i.e., plane and solid) to conduct the QTO process. After unionization at the intersections, the models with default joints, switch joints, and overlaps have the same geometric representation. Then, cutting planes are introduced to cut the integrated solids. Therefore, the same cut solids generate stable results in different modeling scenarios. For example, when computing slab quantities under the same-grade condition, the volume item $A_{slab} \times T_{slab}$ in Eq. (1) is taken care of by the upper solid after cutting the integrated solid with the soffit plane of the slab, instead of multiplying the slab dimensions that vary in the three different models to obtain A_{slab}

and V_{slab}. Further, the results from the BIM authoring software are the same in both same-grade and different-grade situations, which is not aligned with the SMM rules, because the software obtains the quantities based on the 3D geometries without consideration of the semantic data (e.g., concrete grade) carried by the model. In contrast, our method detects the spatial relationships in adjacent elements using the contact detection algorithm so that the contact elements can be identified, and their concrete grade information can be considered in the calculation process to reflect the measurement rules (e.g., situations under $C_{slab} = C_{other}$ and $C_{slab} \neq C_{other}$ in Eq (2)). For the professional QTO software, if the intersections are detected as an overlap area, the reduction rules in SMM will be applied automatically to compute the concrete quantity (see Fig 18 (b)), otherwise, consideration of these intersections would be missed. The existing BIM authoring and professional QTO tools suffer from the limitations of the conventional model-based approach. Through the manipulations of planes and solids instead of dimensions, as well as the considerations of spatial relationships and semantic information, our method is immune to varying BIM model creation methods and can reflect the measurement rules in the calculation.

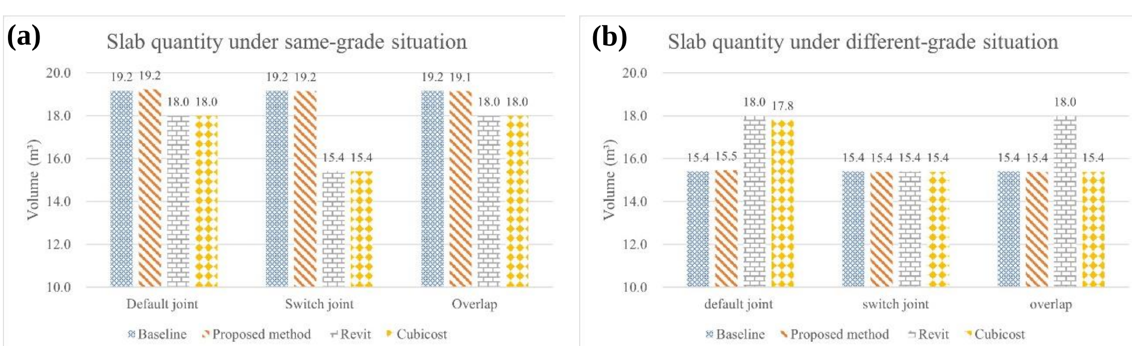
More importantly, the quantities from the proposed method are almost the same as the baseline results in all situations, which means they are reliable and can be used in practice. This is because the proposed method incorporates the measurement rules and utilizes both geometric and semantic information for the calculation. For example, when computing the slab's quantities, the algorithms would use the geometry to obtain the contacted elements, then use the semantics (i.e., concrete grades) to determine the dominance of the slab in Eq. (2). If the concrete grades are the same, the geometry of the intersection parts belongs to the slab's quantities, otherwise, it is not considered as part of the slab, as shown in Fig 18 (a) and (b). However, as explained above, the BIM authoring software simply takes the geometry of the slab and results in inaccuracies. Since there is no overlap at the intersections in the default join and switch join models, the professional QTO software does not apply rule calculations (e.g., reduction, no-reduction) for the intersections and hence has deviations. In addition to SMM rules, another important aspect regarding the accuracy of our proposed method and the deviation of the BIM authoring software and the professional QTO software is the consideration of classification rules. As Fig 21 shows, the element between two walls is modeled as a column. However, it should be classified as a part of the walls instead of the columns, because the width exceed four times its thickness [12]. The corresponding concrete quantities should belong to the wall category. The proposed method incorporates such classification rules in SMM and hence can classify the concrete quantities into the correct categories. However, the BIM authoring software classifies building elements and their quantities purely based on the element categories defined when the model is created. This may lead to incorrect classifications of the building elements and a larger discrepancy in the QTO results.

For the formwork results, as shown in Fig 19, there are no quantities from the BIM authoring software since formworks are not modeled, but our proposed method can provide not only automatic but also almost code-compliant results because it utilizes information from the modeled elements (e.g., soffit, side) and considers the rules comprehensively, with the concept of spatial relationship detection (i.e., contact detection) and manipulations of planes and solids rather than dimensions. For example, when calculating the quantities of formworks to slabs, the item for bottom formwork A_{soffit} in Eq. (17) is obtained by subtracting the soffit areas of the elements in contact with the slabs from the total bottom area of the integrated solid consisting of the slabs and the contacted elements. Instead of taking the areas of the exposed surfaces directly, the algorithms carefully consider the reduction and no-reduction rules at the intersections such as those mentioned in section 3.1.3 to make the results compliant with the measurement standard. For instance, when obtaining the quantities of formworks to columns or walls, the $A_{intersect}$ in Eq. (10) comes from the column-wall or wall-wall intersection areas without consideration of beams through only detecting contacted columns or walls. As presented in Fig 19 (c), the algorithms can also capture the formwork quantities counted in numbers at the cantilever ends (i.e., N_{side} in Eq.(14)). In contrast, although the professional QTO software can also provide almost accurate formwork areas, it cannot output the formwork counted in numbers at cantilever ends directly (see Fig 19 (c)) and misclassifies the formwork to walls into the formwork to columns (see Fig 21), which increases the column formwork and decreases the wall formwork.

In addition, the method to calculate formwork quantities using BIM in [31] was replicated according to the main logic and the results were compared with those from the proposed method. As shown in Table 4, the method in [31] has relatively large deviations about the areas of formworks to columns and walls. The reason is that it directly considers the areas of exposed surfaces as the formwork quantities without careful considerations of reduction and no-reduction rules at the intersections (e.g., no reduction should be conducted for formworks to columns at the beam-column intersections). Furthermore, it does not consider the QTO-specific discrimination rules (e.g., Fig 21) and thus misclassifies the formwork categories and enlarge the deviations. Besides, formworks counted in numbers cannot be captured by [31] since it purely deals with the surface areas.

In summary, our method considers actual complex measurement rules for formworks comprehensively (e.g., considering reduction and no-reduction rules by detection and no-detection of contact elements accordingly, obtaining formwork quantities counted in numbers by using the spatial detection algorithm flexibly, and classifying formwork types appropriately), instead of directly eliminating intersections to obtain exposed areas.

Due to the system error, there are slight differences between the results from our proposed method and the baseline. At the backend, Revit would convert the metric units in the project to imperial ones to store the geometry data. Since in the unionization process, the edges of the elements to be unionized need to align with each other, there is a unit conversion from metric units adopted by the project to the imperial ones in the Dynamo scripts. After calculation, the results would be converted back to standard units. Such unit conversion process and the rounding operation afterwards would lead to small deviations. In addition, the summation would accumulate the deviations in the results. However, the differences are small enough to be acceptable.



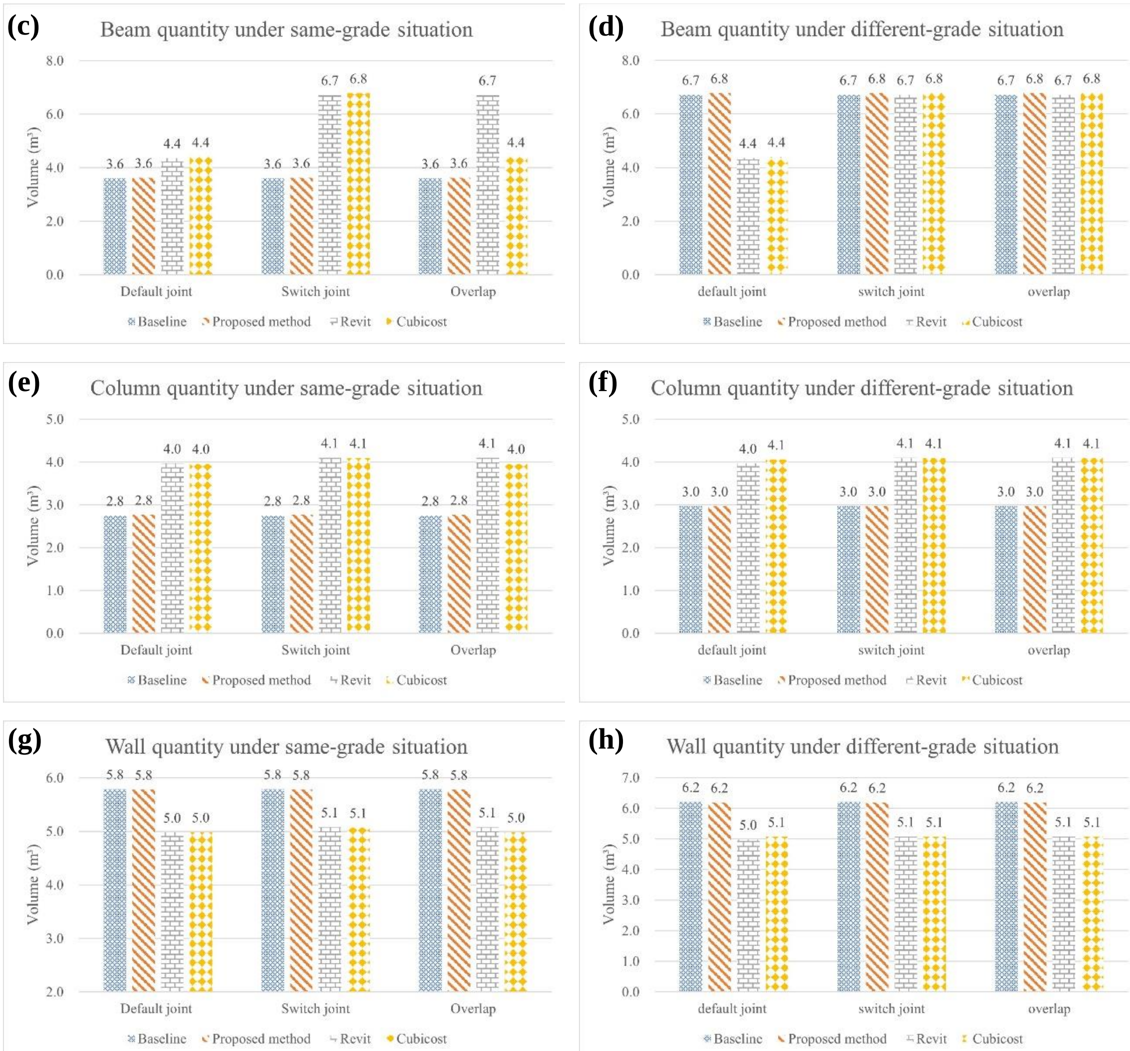
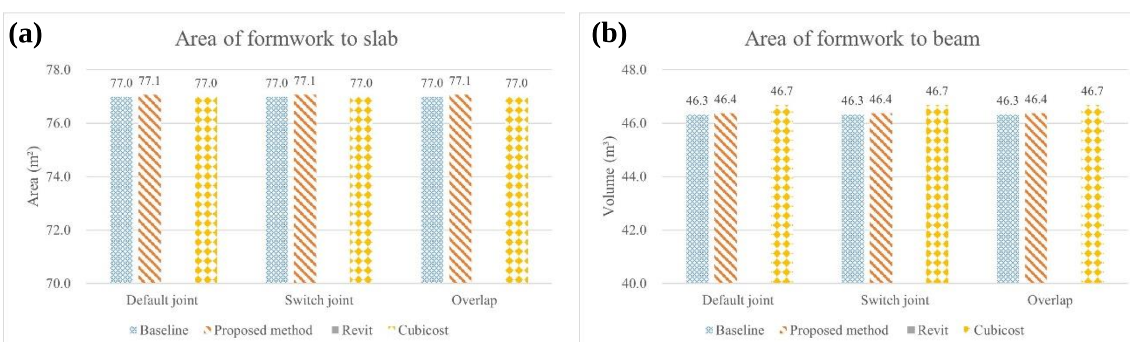


Fig 18. Comparison of concrete quantities



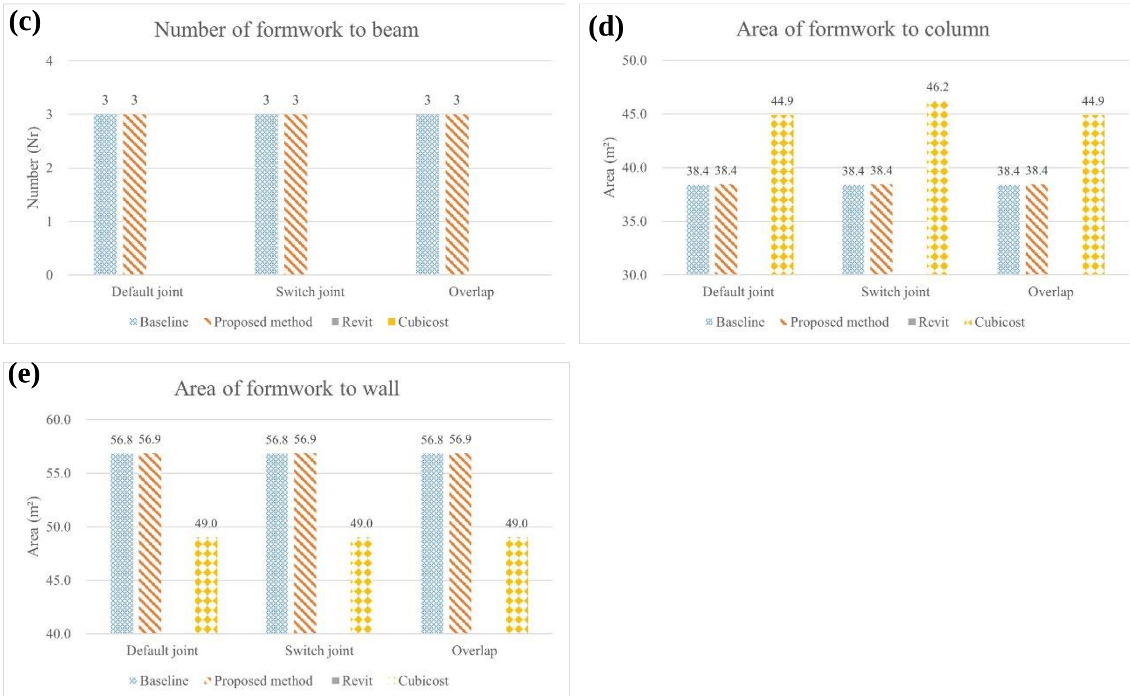
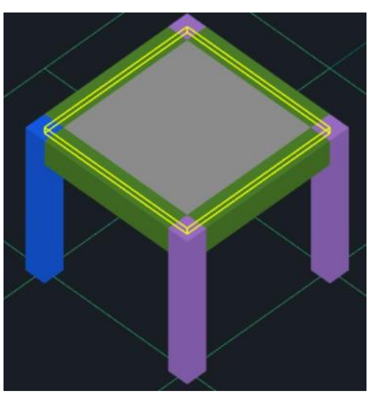


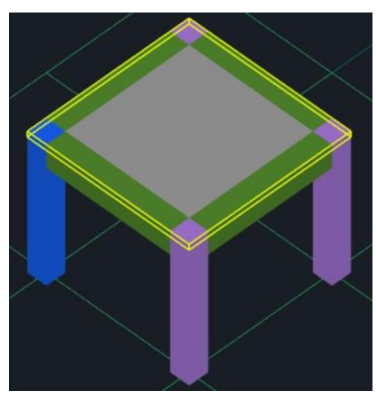
Fig 19. Comparison of formwork quantities

Table 4. Comparison of formwork quantities with [31]

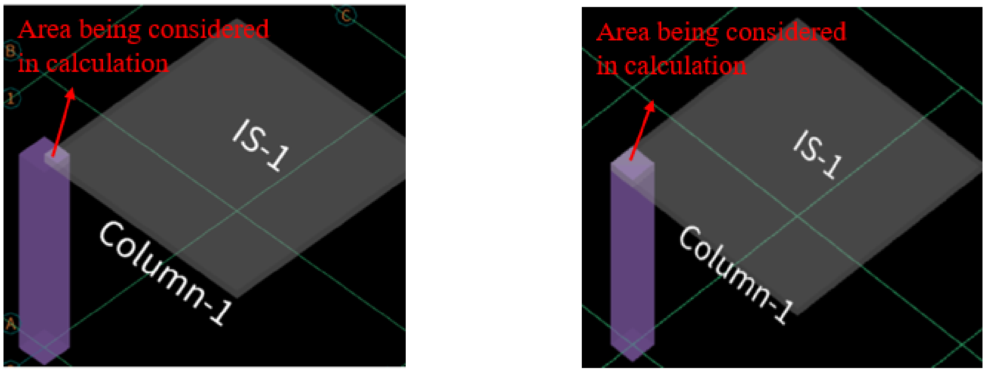
Item	Default joint/Switch joint/Overlap		
	Baseline	Proposed method	[31]
Area of formwork to slab	$77.0m^2$	$77.1m^2$	$77.1m^2$
Area of formwork to beam	$46.3m^2$	$46.4m^2$	$46.7m^2$
Number of formwork to beam	$3\ Nr$	$3\ Nr$	-
Area of formwork to column	$38.4m^2$	$38.4m^2$	$42.3m^2$
Area of formwork to wall	$56.8m^2$	$56.9m^2$	$49.1m^2$



(a) Slab overlaps with one-quarter of the intersection with each column



(b) Slab overlaps with the whole intersection with each column



(c) One quarter of the intersection being considered for reduction/no-reduction

(d) The whole intersection being considered for reduction/no-reduction

Fig 20. Rule calculation at the intersections in Glodon Cubicost TAS

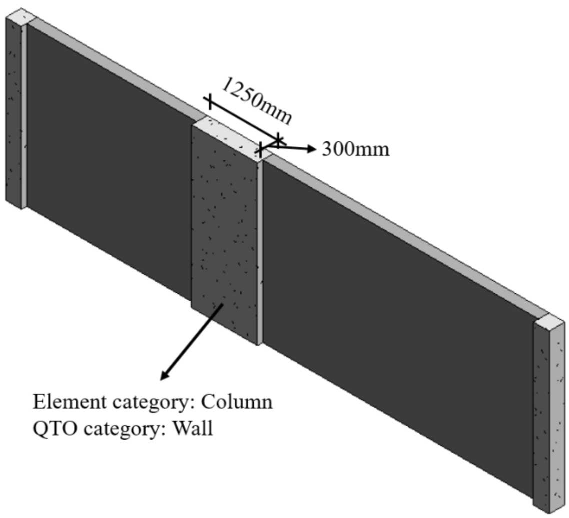


Fig 21. Column and wall discrimination in the QTO process

6. CONCLUSIONS

In this paper, QTO-related information requirements are identified through a typical QTO process map containing the information flow and data exchange among different tasks and stakeholders. A semantic data model for typical in-situ concrete QTO items is established to represent the semantics in the measurement rules. A rule library is also formulated to express the logic for computing the material quantities. The developed data model provides the basis for improving the interoperability and open digital workflow for automatic QTO and/or cost estimation in construction. An automatic semantic auditing method based on linguistic techniques is developed to check the completeness and correctness of QTO-related information such that the information in the BIM models is sufficient for undertaking the QTO process. The principles of the semantic auditing algorithms can be applied not only for QTO but also for other applications to ensure semantic richness. Further, new calculation concepts and methods are designed to utilize both geometric and semantic information to detect the spatial relationships and determine the interactions with measurement rules for automatic and code-compliant quantities in different modeling scenarios, providing quantity surveyors with automation, code-compliance, and robustness in the QTO process.

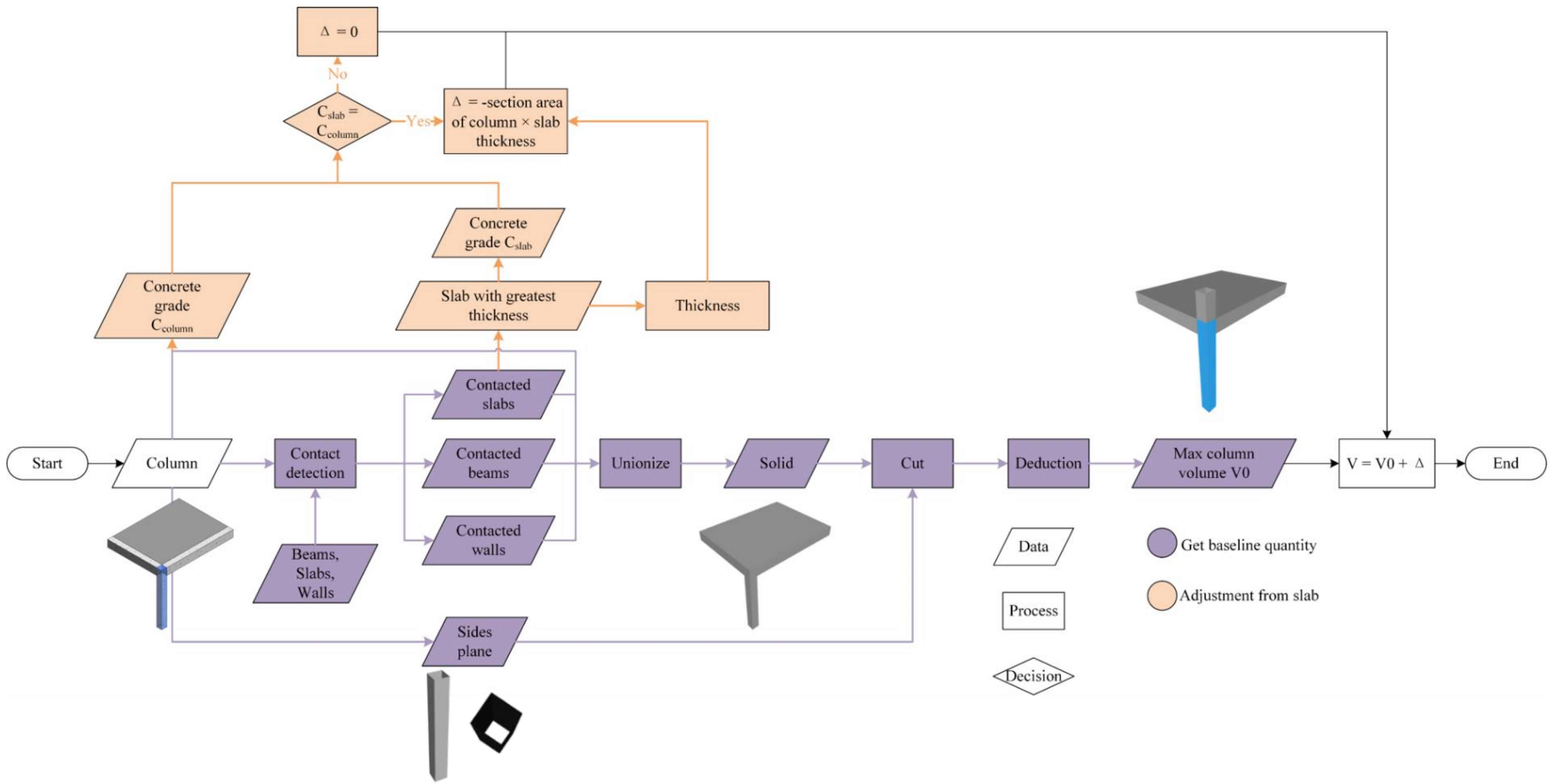
The developed knowledge model-based framework is applied to three illustrative examples. The results of semantic auditing indicate that the algorithms can perform not only keyword matching but also approximate string matching to ensure the quality of the BIM model semantics for QTO purposes. While BIM models can be created by different methods which may cause graphical errors and impact on the calculation of quantities, the proposed framework equipped with the measurement rules and newly designed QTO algorithms can provide automatic, code-compliant, and robust results. Therefore, the knowledge model-based framework presented in this paper establishes an approach for delivering reliable quantities of different QTO items that can be used in practice without adjustments.

The knowledge modeling process in this paper relies on human intervention to extract the QTO-related information (e.g., entities, attributes, relationships). Given that the Natural Language Processing (NLP) and machine learning techniques have the potential to extract pieces of information (e.g., named entities) from documents, automatic information extraction can be explored to support human-free knowledge modeling. In addition, this study covers the typical in-situ concrete elements and formworks, while a wider range of unmodeled items (e.g., finishing) which involves large amounts of semantic information, are not included. Developing more efficient algorithms to learn the rich semantics of unmodeled elements for logical reasoning of the material quantity can be considered as a part of future work.

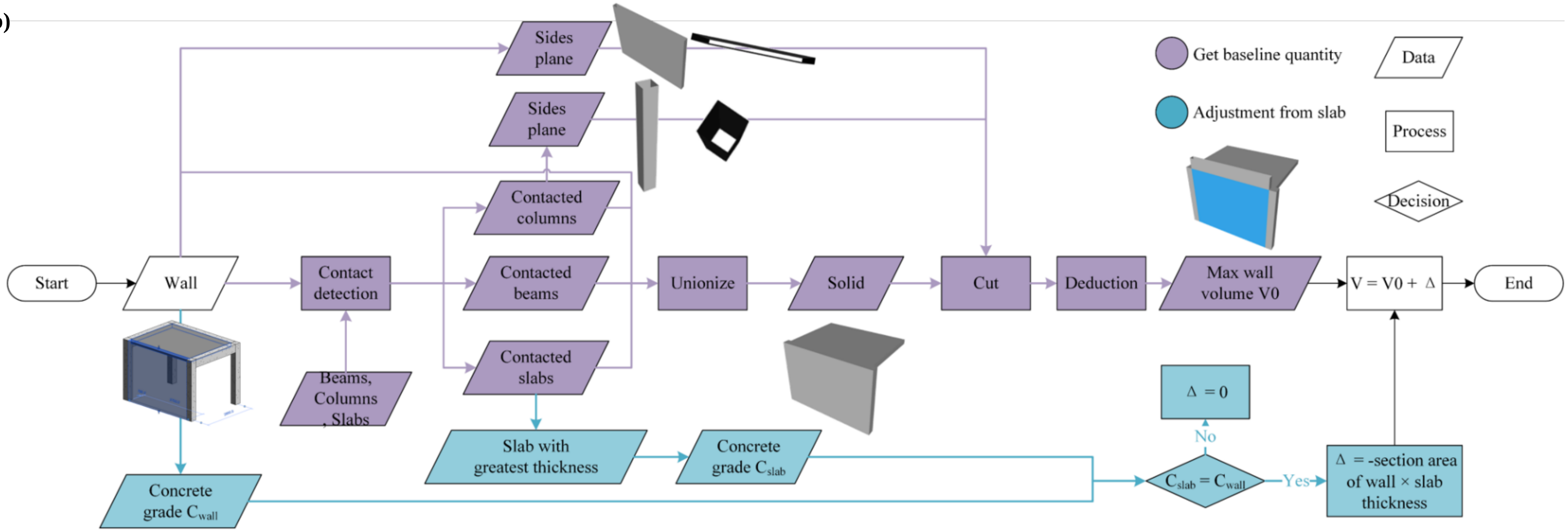
APPENDIX A. Algorithms for measurement of concrete quantities

The computational algorithms for the measurement of columns, walls, and beams are shown in Fig 22. The calculation equation for columns is $V = V_0 + \Delta_{slab}$, where V is the column's quantity, and V_0 is the baseline quantity. In the column's case, it is the maximum possible quantity (i.e., the height of the column is measured up to the top of the slab). Δ_{slab} is the adjustment from the contacted slabs. For walls, the calculation equation is $V = V_0 + \Delta_{slab}$, where V is the wall's quantity, V_0 is the baseline quantity, which is the case that the wall is cut by the contacted columns (i.e., walls are measured between columns) and measured up to the top of the slab. Δ_{slab} is the adjustment from contacted slabs. For beams, the calculation equation is $V = V_0 + \Delta_{slab}$, where V is the column's quantity, and V_0 is the baseline quantity. In the beam's case, it is the minimum possible quantity, which is the case that the beam is cut by the contacted columns and walls (i.e., beams are measured between columns and walls) and then further cut by the soffit plane of the contacted slab (i.e., measured up to the soffit of the slab). Δ_{slab} is the adjustment from the contacted slabs.

(a)


Δ = 0
No
C_{slab} = C_{column}
Yes
Δ = -section area of column × slab thickness
Concrete grade C_{slab}
Concrete grade C_{column}
Slab with greatest thickness
Thickness
Contacted slabs
Start
Column
Contact detection
Contacted beams
Unionize
Solid
Cut
Deduction
Max column volume V0
V = V0 + Δ
End
Contacted walls
Beams, Slabs, Walls
Sides plane
Data
Process
Decision
Get baseline quantity
Adjustment from slab

(b)


Sides plane
Sides plane
Get baseline quantity
Adjustment from slab
Data
Process
Decision
Contacted columns
Start
Wall
Contact detection
Contacted beams
Unionize
Solid
Cut
Deduction
Max wall volume V0
V = V0 + Δ
End
Beams, Columns , Slabs
Contacted slabs
Δ = 0
No
Slab with greatest thickness
Concrete grade C_{slab}
$C_{slab} = C_{wall}$
Yes
Δ = -section area of wall × slab thickness
Concrete grade C_{wall}

(c)

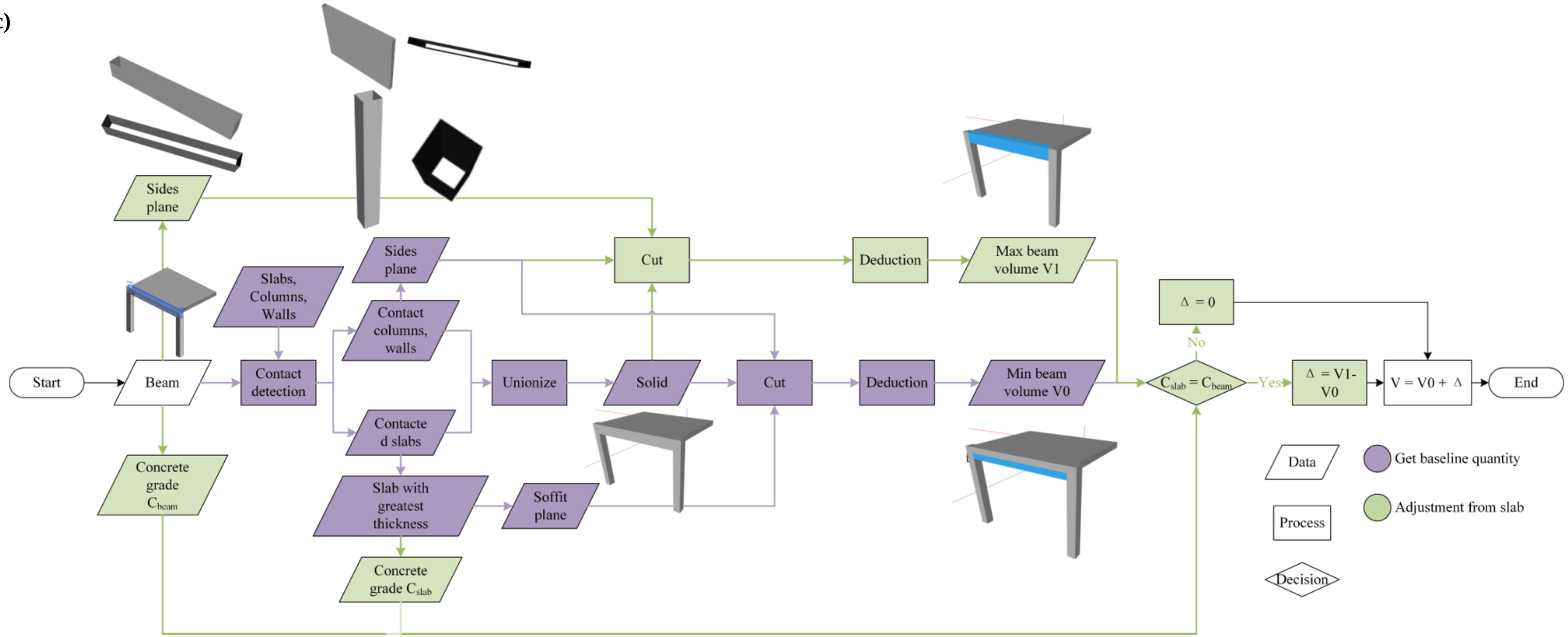


Fig 22. Algorithms for measurement of concrete quantities (Fig 22 (a): measurement of column quantities; Fig 22 (b): measurement of wall quantities; Fig 22 (c): measurement of beam quantities)

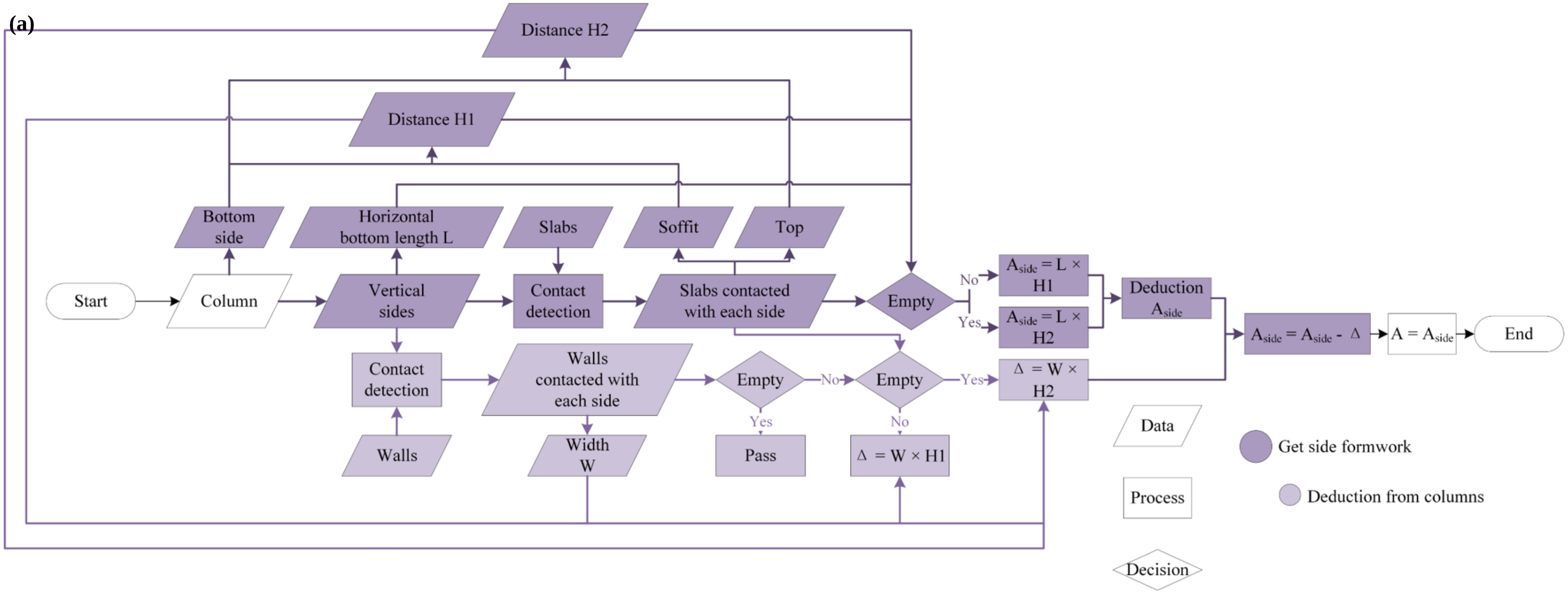
APPENDIX B. Algorithms for measurement of formwork quantities

The computational algorithms for the measurement of formworks to columns, walls, and slabs are shown in Fig 23. The calculation equation for the formwork to columns is $A = A_{side}$, where A is the formwork's quantity, A_{side} is the quantity of the formwork to the column sides. If there are identified intersected walls with column sides, the intersection areas should be subtracted, which are calculated up to the soffit or the top of the slabs, while the areas of the column-beam intersections should not be deducted. Finally, the quantities of formwork to the column equal the side areas after subtracting the areas of the column-wall intersections.

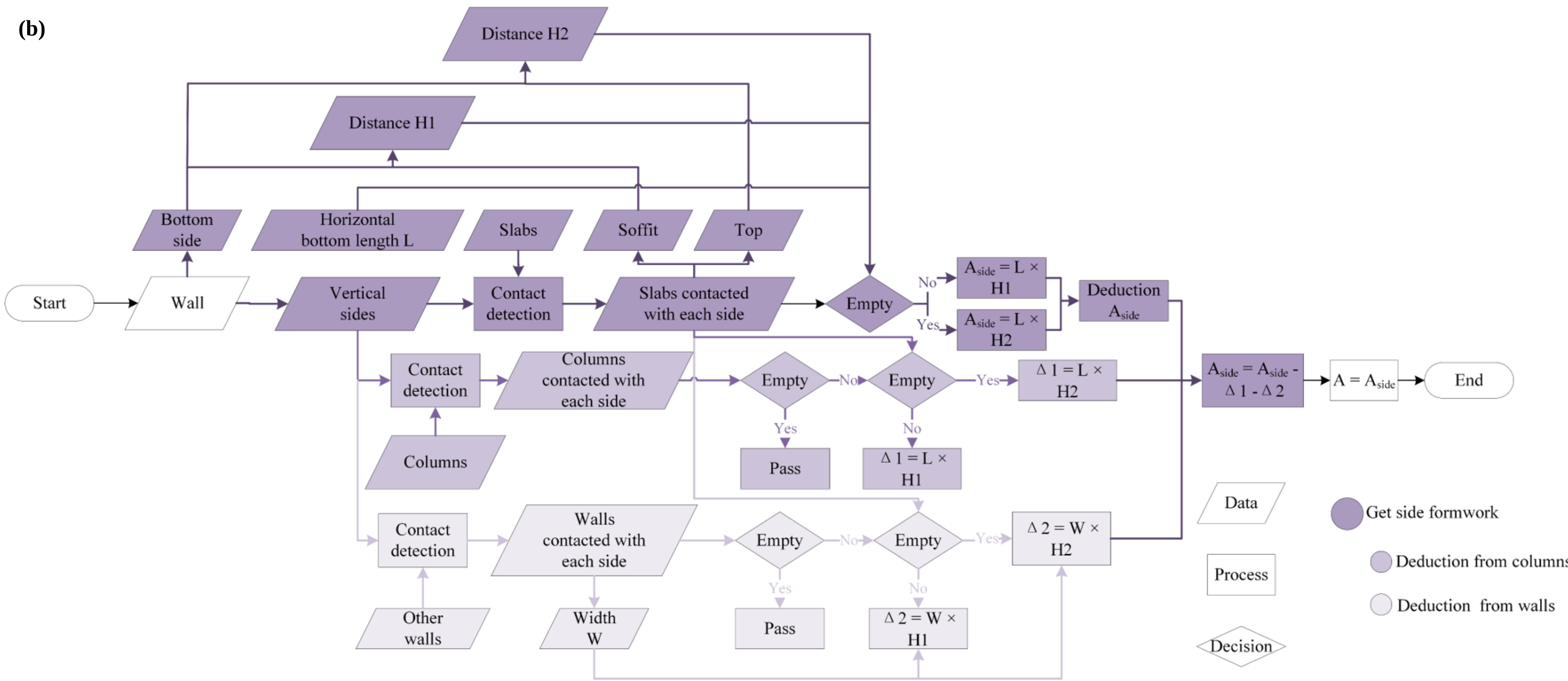
The calculation equation for the formwork to walls is $A = A_{side}$, where A is the formwork's quantity, and A_{side} is the quantity of the formwork to the wall sides. The calculations are similar to columns except for one additional reduction from the wall-wall intersections.

The calculation equation for formwork to slabs is $A = A_{side} + A_{bottom/top}$, where A is the formwork's quantity, A_{side} is the quantity of the formwork to the slab edges, $A_{bottom/top}$ is the quantity of the formwork to the slab's bottom and top. The total slab area is first obtained through the cutting of the solids unionized from the slabs and their contacted elements. Following this, A_{bottom} is obtained after subtraction of the reduction items and the areas of the soffits or bottoms of the contacted beams, columns and walls detected earlier. On the other hand, the algorithm extracts the vertical sides to find contacted elements. If the side has no contacted elements, it is an edge that needs formwork and its area is counted into the A_{side}.

(a)


Distance H2
Distance H1
Bottom side
Horizontal bottom length L
Slabs
Soffit
Top
Start
Column
Vertical sides
Contact detection
Slabs contacted with each side
Empty
No
Yes
A_side = L × H1
A_side = L × H2
Deduction A_side
A_side = A_side - Δ
A = A_side
End
Contact detection
Walls contacted with each side
Empty
No
Empty
Yes
Δ = W × H2
Walls
Width W
Yes
Pass
No
Δ = W × H1
Data
Process
Decision
Get side formwork
Deduction from columns

(b)


Distance H2
Distance H1
Bottom side
Horizontal bottom length L
Slabs
Soffit
Top
Start
Wall
Vertical sides
Contact detection
Slabs contacted with each side
Empty
No
Yes
A_{side} = L × H1
A_{side} = L × H2
Deduction A_{side}
Contact detection
Columns contacted with each side
Columns
Empty
No
Empty
Yes
Δ 1 = L × H2
Yes
Pass
No
Δ 1 = L × H1
A_{side} = A_{side} - Δ 1 - Δ 2
A = A_{side}
End
Contact detection
Walls contacted with each side
Other walls
Width W
Empty
No
Empty
Yes
Δ 2 = W × H2
Yes
Pass
No
Δ 2 = W × H1
Data
Process
Decision
Get side formwork
Deduction from columns
Deduction from walls

(c)

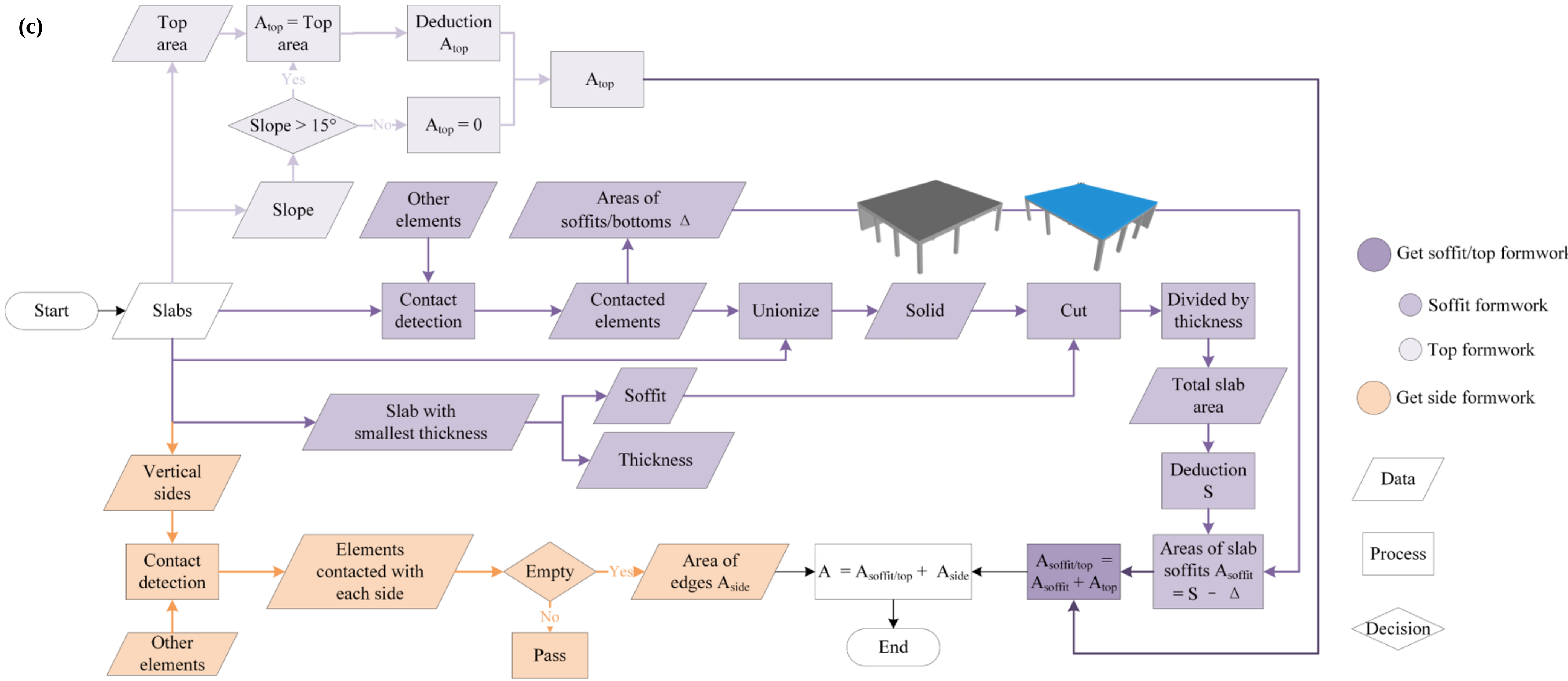


Fig 23. Algorithms for measurement of formwork quantities (Fig 23 (a): measurement of formwork to column; Fig 23 (b): measurement of formwork to wall; Fig 23 (c): measurement of formwork to slab)

REFERENCES

[1] S. Aram, C. Eastman, R. Sacks, A knowledge-based framework for quantity takeoff and cost estimation in the AEC industry using BIM, in: The 31st International Symposium on Automation and Robotics in Construction and Mining (ISARC), Sydney, Australia, 2014: pp. 434–442. https://doi.org/10.22260/ISARC2014/0058.

[2] C. Khosakitchalert, N. Yabuki, T. Fukuda, Improving the accuracy of BIM-based quantity takeoff for compound elements, Automation in Construction. 106 (2019) pp. 102891. https://doi.org/10.1016/j.autcon.2019.102891.

[3] A. Monteiro, J.P. Martins, A survey on modeling guidelines for quantity takeoff-oriented BIM-based design, Automation in Construction. 35 (2013) pp. 238–253. https://doi.org/10.1016/j.autcon.2013.05.005.

[4] L. Holm, J. E. Schaufelberger, D. Griffin, T. Cole, Construction cost estimating: process and practices, 1st Edition, Pearson, 978-0130496652, 2004.

[5] R. Sacks, C. Eastman, G. Lee, P. Teicholz, BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers, John Wiley & Sons, 978-1-119-28753-7, 2018.

[6] M. Juszczyk, R. Kozik, A. Leśniak, E. Plebankiewicz, K. Zima, Errors in the Preparation of Design Documentation in Public Procurement in Poland, Procedia Engineering. 85 (2014) pp. 283–292. https://doi.org/10.1016/j.proeng.2014.10.553.

[7] A. Sattineni, R.H. Bradford, Estimating with BIM: A Survey of US Construction Companies, in: The 28th International Symposium on Automation and Robotics in Construction (ISARC), Seoul, Korea, 2011: pp. 564–569. https://doi.org/10.22260/ISARC2011/0103.

[8] A. Nadeem, A.K.D. Wong, F.K.W. Wong, Bill of Quantities with 3D Views Using Building Information Modeling, Arabian Journal for Science and Engineering. 40 (2015) pp. 2465–2477. https://doi.org/10.1007/s13369-015-1657-2.

[9] C. Khosakitchalert, N. Yabuki, T. Fukuda, Automated modification of compound elements for accurate BIM-based quantity takeoff, Automation in Construction. 113 (2020) pp. 103142. https://doi.org/10.1016/j.autcon.2020.103142.

[10] S. Kim, S. Chin, S. Kwon, A Discrepancy Analysis of BIM-Based Quantity Take-Off for Building Interior Components, Journal of Management in Engineering. 35 (2019) pp. 05019001. https://doi.org/10.1061/(ASCE)ME.1943-5479.0000684.

[11] B. Yang, B. Zhang, J. Wu, B. Liu, Z. Wang, A BIM-based Quantity Calculation Framework for Frame-shear Wall Structure, Structural Engineering International. 29 (2019) pp. 282–291. https://doi.org/10.1080/10168664.2018.1550352.

[12] Hong Kong Institute of Surveyors, Hong Kong Standard Method of Measurement of Building Works, Fourth Edition, Pace Publishing Limited, 988-98402-4-3, 2005.

[13] J. Choi, H. Kim, I. Kim, Open BIM-based quantity take-off system for schematic estimation of building frame in early design stage, Journal of Computational Design and Engineering. 2 (2015) pp. 16–25. https://doi.org/10.1016/j.jcde.2014.11.002.

[14] O.A. Olatunji, W. Sher, N. Gu, Building Information Modeling and Quantity Surveying Practice, Emirates Journal for Engineering Research. 15 (2010) pp. 67–70.

[15] Z. Ma, Z. Wei, X. Zhang, Semi-automatic and specification-compliant cost estimation for tendering of building projects based on IFC data of design model, Automation in Construction. 30 (2013) pp. 126–135. https://doi.org/10.1016/j.autcon.2012.11.020.

[16] Glodon, Cubicost TAS, (2021). https://cubicost.com/ (accessed April 19, 2021).

[17] M.N. Jadid, M.M. Idrees, Cost estimation of structural skeleton using an interactive automation algorithm: A conceptual approach, Automation in Construction. 16 (2007) pp. 797–805. https://doi.org/10.1016/j.autcon.2007.02.007.

[18] R. Bečvarovská, P. Matějka, Comparative Analysis of Creating Traditional Quantity Takeoff Method and Using a BIM Tool, in: Construction Maeconomics Conference 2014, 2014.

http://www.conference-cm.com/podklady/history5/Prispevky/paper_becvarovska.pdf (accessed November 30, 2020).

[19] F.K.T. Cheung, J. Rihan, J. Tah, D. Duce, E. Kurul, Early stage multi-level cost estimation for schematic BIM models, Automation in Construction. 27 (2012) pp. 67–77. https://doi.org/10.1016/j.autcon.2012.05.008.

[20] A. Fazeli, M.S. Dashti, F. Jalaei, M. Khanzadi, An integrated BIM-based approach for cost estimation in construction projects, Engineering, Construction and Architectural Management. ahead-of-print (2020). https://doi.org/10.1108/ECAM-01-2020-0027.

[21] S.-K. Lee, K.-R. Kim, J.-H. Yu, BIM and ontology-based approach for building cost estimation, Automation in Construction. 41 (2014) pp. 96–105. https://doi.org/10.1016/j.autcon.2013.10.020.

[22] J.H. Yousif, S. N. Abdul Majeed, F. J. I. Al Azzawi, Web-Based Architecture for Automating Quantity Surveying Construction Cost Calculation, Infrastructures. 5 (2020) 45. https://doi.org/10.3390/infrastructures5060045.

[23] J. Franco, F. Mahdi, H. Abaza, Using Building Information Modeling (BIM) for Estimating and Scheduling, Adoption Barriers, Universal Journal of Management. 3 (2015) 376–384. https://doi.org/10.13189/ujm.2015.030905.

[24] F. Leite, A. Akcamete, B. Akinci, G. Atasoy, S. Kiziltas, Analysis of modeling effort and impact of different levels of detail in building information models, Automation in Construction. 20 (2011) 601–609. https://doi.org/10.1016/j.autcon.2010.11.027.

[25] S.-A. Kim, S. Chin, S.-W. Yoon, T.-H. Shin, Y.-S. Kim, C. Choi, Automated Building Information Modeling System for Building Interior to Improve Productivity of BIM-based Quantity Take-Off, in: The 26th International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Austin, USA, 2009: pp. 492–496. https://doi.org/10.22260/ISARC2009/0017.

[26] C.E. Firat, D. Arditi, J.-P. Hämäläinen, J. Stenstrand, J. Kiiras, Quantity take-off in model-based systems, in: Proceedings of the 27th CIB W78 International Conference, Cairo, Egypt, 2010: pp. 16–18. https://itc.scix.net/pdfs/w78-2010-112.pdf (accessed April 5, 2021).

[27] K. Zima, Impact of information included in the BIM on preparation of Bill of Quantities, Procedia Engineering. 208 (2017) pp. 203–210. https://doi.org/10.1016/j.proeng.2017.11.039.

[28] J. Cho, J. Chun, Cost Estimating Methods for RC Structures by Quantity Takeoff and Quantity Prediction in the Design Development Stage, Journal of Asian Architecture and Building Engineering. 14 (2015) pp. 65–72. https://doi.org/10.3130/jaabe.14.65.

[29] M. Rajabi, T. Bigga, M.A. Bartl, Optimization of the Quantity Take-off (QTO) Process for Mechanical, Electrical and Plumbing (MEP) Trades in Tender Estimation Phase of the Construction Projects, in: The 32nd International Symposium on Automation and Robotics in Construction (ISARC), IAARC, Oulu, Finland, 2015: pp. 1–8. https://doi.org/10.22260/ISARC2015/0022.

[30] C. Lim, W.-K. Hong, D. Lee, S. Kim, Automatic rebar estimation algorithms for integrated project delivery, Journal of Asian Architecture and Building Engineering. 15 (2016) pp. 411–418. https://doi.org/10.3130/jaabe.15.411.

[31] C. Khosakitchalert, N. Yabuki, T. Fukuda, Automatic Concrete Formwork Quantity Takeoff Using Building Information Modeling, in: Proceedings of the 19th International Conference on Construction Applications of Virtual Reality (CONVR2019), Bangkok, Thailand, 2019: pp. 21–28. https://www.researchgate.net/profile/Chavanont-Khosakitchalert/publication/337286106_Automatic_Concrete_Formwork_Quantity_Takeoff_Using_Building_Information_Modeling/links/5e51e8c4299bf1cdb94006ff/Automatic-Concrete-Formwork-Quantity-Takeoff-Using-Building-Information-Modeling.pdf (accessed December 15, 2020).

[32] Y. Çepni, A. Akcamete, Automated BIM-based Formwork Quantity Take-Off, in: Proceedings of the 20th International Conference on Construction Applications of Virtual Reality (CONVR2019), 2020: pp. 220–227. https://www.researchgate.net/profile/Asli-Akcamete/publication/345896054_Automated_BIM-based_Formwork_Quantity_Take-Off/links/5fb1199945851518fda6e703/Automated-BIM-based-Formwork-Quantity-Take-Off.pdf (accessed December 20, 2020).
[33] R. Romanovskyi, L.S. Mejia, E.R. Azar, BIM-based Decision Support System for Concrete Formwork Design, in: The 36st International Symposium on Automation and Robotics in Construction (ISARC), Banff Alberta, Canada, 2019: pp. 1129–1135. https://doi.org/10.22260/ISARC2019/0150.
[34] C. Hyun, C. Jin, Z. Shen, H. Kim, Automated optimization of formwork design through spatial analysis in building information modeling, Automation in Construction. 95 (2018) pp. 193–205. https://doi.org/10.1016/j.autcon.2018.07.023.
[35] J. Lee, Y.-J. Park, C.-H. Choi, C.-H. Han, BIM-assisted labor productivity measurement method for structural formwork, Automation in Construction. 84 (2017) pp. 121–132. https://doi.org/10.1016/j.autcon.2017.08.009.
[36] H. Wei, S. Zheng, L. Zhao, R. Huang, BIM-based method calculation of auxiliary materials required in housing construction, Automation in Construction. 78 (2017) pp. 62–82. https://doi.org/10.1016/j.autcon.2017.01.022.
[37] K. Kim, J. Teizer, Automatic design and planning of scaffolding systems using building information modeling, Advanced Engineering Informatics. 28 (2014) pp. 66–80. https://doi.org/10.1016/j.aei.2013.12.002.
[38] V. Peansupap, S. Thuanthongdee, Levels of Development in BIM for Supporting Cost Estimation of Building Construction Projects, in: The 16th International Conference on Computing in Civil and Building Engineering (ICCCBE), Osaka, Japan, 2016: pp. 671–678. http://www.see.eng.osaka-u.ac.jp/seeit/icccbe2016/Proceedings/Full_Papers/085-346.pdf (accessed December 29, 2020).
[39] L. Andersson, K. Farrell, O. Moshkovich, C. Cranbourne, Implementing Virtual Design and Construction using BIM: Current and future practices, Routledge, 9781138019942, 2016.
[40] B. Hardin, D. McCool, BIM and Construction Management: Proven Tools, Methods, and Workflows, 2nd Edition, John Wiley & Sons, 978-1-118-94276-5, 2015.
[41] T. Akanbi, J. Zhang, Y.-C. Lee, Data-Driven Reverse Engineering Algorithm Development Method for Developing Interoperable Quantity Takeoff Algorithms Using IFC-Based BIM, Journal of Computing in Civil Engineering. 34 (2020) pp. 04020036. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000909.
[42] H. Liu, M. Lu, M. Al-Hussein, Ontology-based semantic approach for construction-oriented quantity take-off from BIM models in the light-frame building industry, Advanced Engineering Informatics. 30 (2016) pp. 190–207. https://doi.org/10.1016/j.aei.2016.03.001.
[43] C. Khosakitchalert, N. Yabuki, T. Fukuda, Development of BIM-based quantity takeoff for light-gauge steel wall framing systems, Journal of Information Technology in Construction (ITcon). 25 (2020) pp. 522–544. https://doi.org/10.36680/j.itcon.2020.030.
[44] Z. Ma, Z. Liu, Z. Wei, Formalized Representation of Specifications for Construction Cost Estimation by Using Ontology, Computer-Aided Civil and Infrastructure Engineering. 31 (2016) pp. 4–17. https://doi.org/10.1111/mice.12175.
[45] S. Xu, K. Liu, L.C.M. Tang, W. Li, A framework for integrating syntax, semantics and pragmatics for computer-aided professional practice: With application of costing in construction industry, Computers in Industry. 83 (2016) pp. 28–45. https://doi.org/10.1016/j.compind.2016.08.004.
[46] F.H. Abanda, B. Kamsu-Foguem, J.H.M. Tah, BIM – New rules of measurement ontology for construction cost estimation, Engineering Science and Technology, an International Journal. 20 (2017) pp. 443–459. https://doi.org/10.1016/j.jestch.2017.01.007.

[47] buildingSMART, IFC4_ADD2_TC1 - 4.0.2.1 [Official], (2020). https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/ (accessed August 17, 2021).

[48] V. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Soviet Physics Doklady. 10 (1966) pp. 707–710. https://nymity.ch/sybilhunting/pdf/Levenshtein1966a.pdf (accessed December 22, 2020).

[49] Dynamo, Open source graphical programming for design, 2020. https://dynamobim.org/ (accessed January 10, 2021).

[50] PyEnchant, pyenchant: Python bindings for the Enchant spellchecker, 2020. https://github.com/pyenchant/pyenchant (accessed December 23, 2020).