

Developing a Knowledge-based System for Semantic Enrichment and Automatic BIM-based Quantity Take-off

Hao Liu^a, Jack C.P. Cheng^a and Vincent J.L. Gan^b

^aDepartment of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR

^bDepartment of Building, School of Design and Environment, National University of Singapore, Singapore

E-mail: hliuci@conncet.ust.hk, cejcheng@ust.hk, vincent.gan@nus.edu.sg

Abstract –

In construction cost estimation, building information modelling (BIM) has been commonly utilized to support automatic quantity take-off (QTO). However, conventional BIM models do not contain all the necessary information for QTO, and the calculation does not follow the descriptive rules in standard method of measurement (SMM), which impact the cost estimation accuracy. Therefore, this paper presents a new data model and knowledge-based system to incorporate the required SMM rules, which greatly facilitates BIM software in automatic and accurate QTO. The proposed new methods involve the development of a generic data model by identifying and incorporating the required information (e.g., geometry, semantics) in SMM. Following this, information checking algorithms are developed to check the information completeness and textural errors in QTO practices. Furthermore, the descriptive rules in SMM are defined to create a knowledge library that guides the BIM software in performing automatic QTO. Results of illustrative examples indicate that the proposed new methods can accurately compute the quantities of building components in compliant with SMM, regardless of different approaches for model creation. The proposed methods also linguistically identify the textural errors of parameters and check the compliance of descriptive rules for better QTO practices. Practitioners can automate the BIM-based QTO process to reduce the inaccuracies, time, and errors of cost estimation.

Keywords –

Building information modeling (BIM); Quantity take-off (QTO); Knowledge model; Model auditing; SMM-compliance

1 Introduction

Quantity take-off (QTO) is a critical process in

construction projects. The measured quantities are the prerequisites to many subsequent tasks such as cost estimation, material purchase and production planning [1]. The advent of Building Information Modelling (BIM) technology has brought revolutionary impacts on QTO as quantities are extracted directly from the 3D design models which contain geometry and semantic attributes of components. This makes the QTO process faster and more reliable than the conventional methods [2]. However, the automation and accuracy of QTO is still matter of concern as BIM models do not incorporate all the necessary information for conducting an accurate QTO [3], and the calculation of quantities does not follow the descriptive rules in the standard method of measurement (SMM), causing potential inaccuracies and errors.

Therefore, the objective of this paper is to develop a new knowledge-based system to incorporate all the necessary information and SMM descriptive rules for providing automatic and accurate QTO. Through the identification of information requirements and rules definitions from SMM, a generic data model is first developed to represent the 3D geometry and semantic attributes of buildings. Following this, information checking algorithms are developed to extract and linguistically check the completeness and correctness of required information to perform the QTO. Furthermore, the descriptive rules in SMM are mathematically formulated to create a knowledge library, which guides the BIM software in automatic QTO. The proposed methodology is validated through illustrative examples. This research contributes a new approach to check the required information and textural errors as well as to calculate accurate quantities in accordance with SMM for inaccurate BIM models. Another main contribution is the establishment of a generalized approach to enhance the automation, accuracy, and efficiency of QTO.

2 Literature Review

Early research on QTO attempted to link the 2D CAD drawings with the bill of quantities automatically to reduce human errors as far as possible [4]. BIM provides the ability to extract quantities from design 3D model directly and therefore is faster and more reliable than traditional methods that are based on 2D drawings. Through a case study comparing BIM-based QTO and traditional methods, Bečvarovská and Matějka [5] concluded that time is saved and errors are reduced after applying BIM. In addition to those extracting quantities from BIM models directly, research in recent years emphasized more on problems of inadequate information in BIM models and rule-compliance of the results. Choi et al. [6] proposed an open BIM-based approach to calculate quantities of structural framing, which would check whether the BIM model contains adequate information before performing QTO. Liu et al. [7] developed an ontology-based system to infer non-expressed information in BIM models to support QTO for wall framing. By studying the New Rules of Measurement (NRM), Abanda et al. [8] developed and used ontology to calculate quantities in accordance with the NRM from BIM models.

While the previous studies have explored BIM-based QTO and the corresponding information interoperability problem, there is still a lack of a generic data model to represent the essential 3D geometry and semantic attributes of buildings to perform automatic QTO. In addition, the inaccuracy of BIM-based QTO due to the lack of SMM descriptive rules is still a matter of concern. Besides, unintended errors in geometric and/or semantic information of design BIM models is another problem, which impacts the recognition of parameters in performing an accurate QTO. Therefore, it is of paramount importance to develop a new intelligent and automatic approach for BIM-based QTO.

3 Methodology

Figure 1 shows the methodology framework for the knowledge-based system. Since QTO is a knowledge-

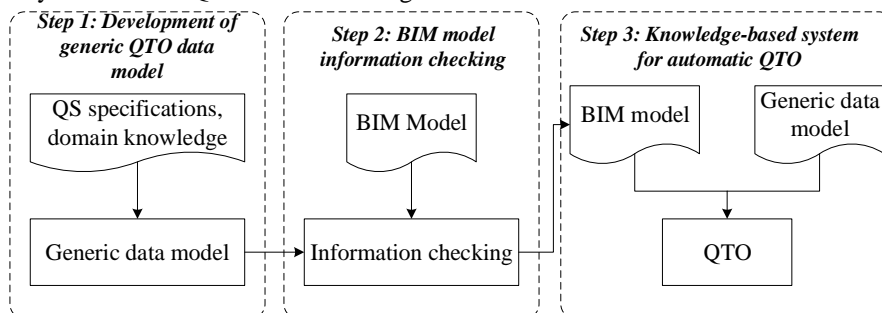


Figure 1. Proposed knowledge-based system

intensive process [9], the first step is to leverage SMM and domain knowledge from quantity surveyors to build a generic data model which incorporates all the essential QTO-related information. To ensure that BIM models contain adequate and correct semantics to extract quantities, the information checking step is needed before the QTO step, where an information checking method is developed to linguistically check the BIM model information based on the data model. Once the QTO-related information is provided and the BIM models have been checked, the descriptive SMM rules are mathematically formulated in a knowledge library to support the computation of building quantities.

3.1 Development of the generic data model

In this step, the information requirements from the measurement standard for performing automatic QTO, including QTO-related entities, attributes, and the semantic relations among them, are defined. This study adopts the Hong Kong Standard Method of Measurement of Building Works – Fourth Edition [10] as the information source for the generic data model.

Unified Modelling Language (UML) diagrams are adopted as the tools for structuring the data model. Figure 2 shows a data model for taking off quantities of some typical in-situ concrete elements as an example, using UML class diagrams. Beam and suspended slab are two specific components, so here the *In-SituConcreteComponent* class has such two sub-classes, which inherit the attributes from it and have their own attributes to meet the information requirements of different components. Each component may have voids and have different shapes with different dimensions (e.g., rectangular, circle, polygon). The calculations of the beam and suspended slab involve the concrete grade information from each other, so they are associated. A typical beam-suspended slab joint is chosen as an example to further elaborate the measurement rules. As shown in Figure 3, the computation flow is structured in an UML activity diagram, showing the calculation priorities between beams and suspended slabs under different situations in terms of concrete grades.

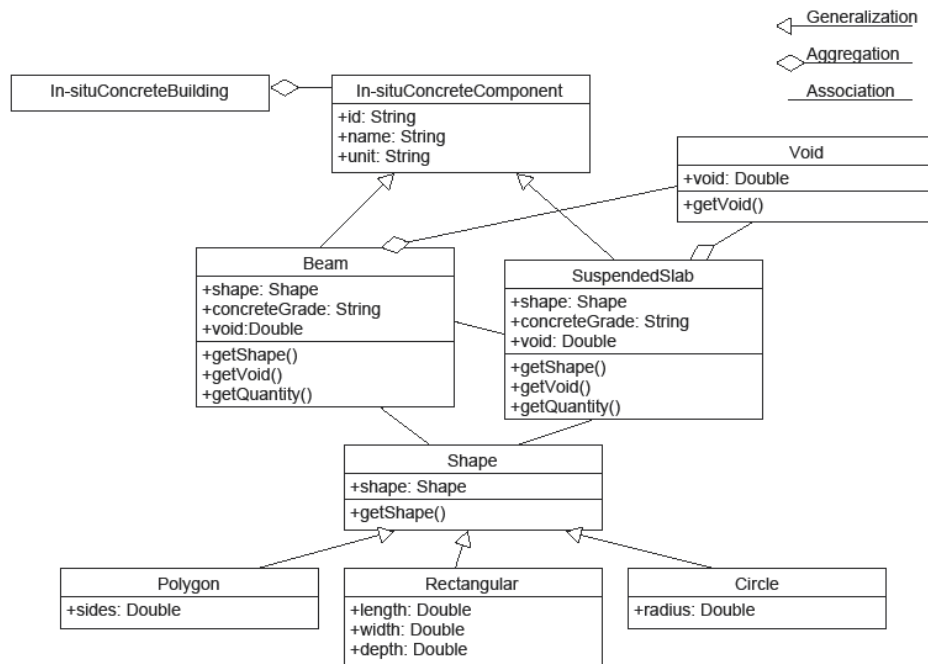


Figure 2. Data model for QTO in typical in-situ concrete beams and suspended slabs

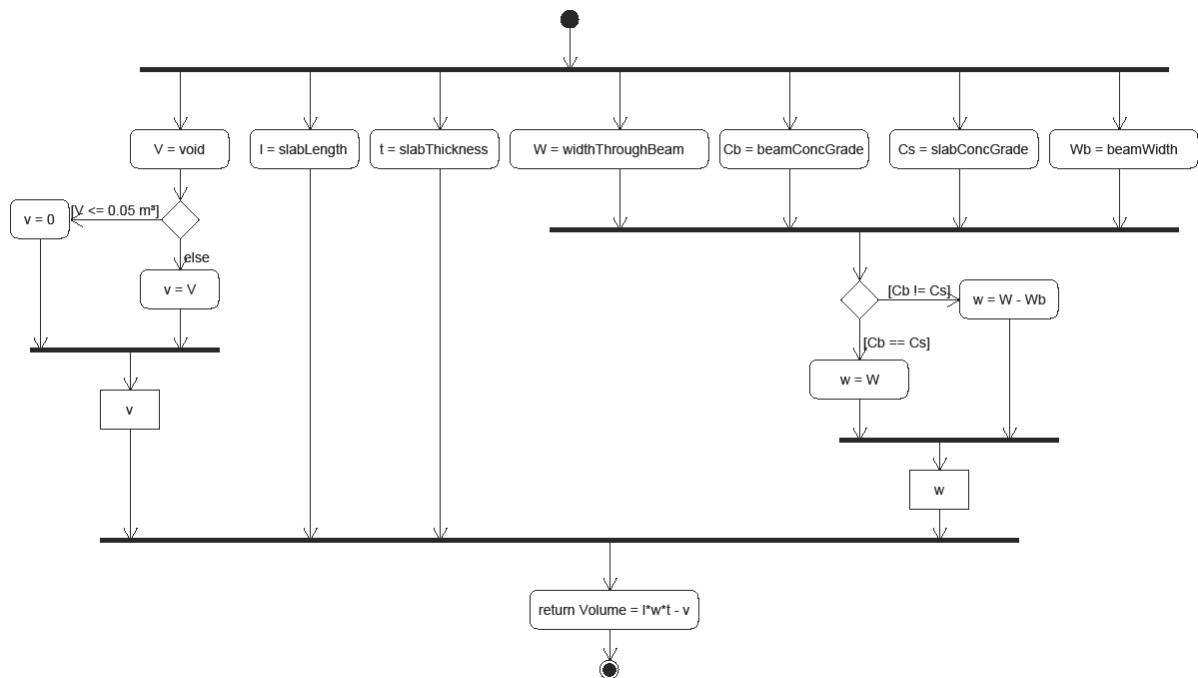


Figure 3. UML activity diagram for measurement of typical beam-suspended slab joint

3.2 BIM Model Information Checking

Then, algorithms are developed to perform the model information checking for QTO, as shown in Figure 4. The semantic attributes are extracted from

design BIM models. The computerized algorithm can automatically check and identify unintended textual errors, and informs users to input more data if the BIM model does not contain adequate information for QTO.

To check the information completeness, design BIM

model information is first extracted. Then, null values and the associated parameters are removed before matching parameter names with the required information derived from the generic data model to filter parameters with names existed but values missed. The required information is concluded as existed 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. Levenshtein distance is a string metric measuring the similarity between two strings [11]. 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 the two compared strings are. This setting enables a fussy recognition so that the algorithm would 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 case, the algorithm can still recognize the existence of this parameter and would not report that it is missed, which reflects the fact of the model.

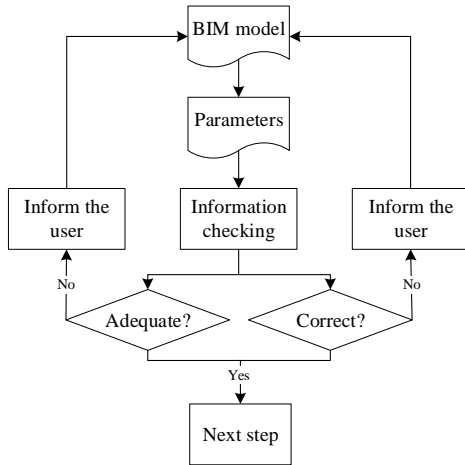


Figure 4. BIM Model Information Checking

Details of the information correctness checking and auto-correction of textural errors are shown in Figure 8. First, the textural parameters are separated into multiple words by a separator list that contains common punctuations such as underline, dash, and space. The separators from the input are kept in a separator container for re-joining the words later. Then, a natural language processing (NLP) checker equipped with different customized domain dictionaries are used to check the correctness of each word. If the result is false, the algorithm will correct the word if the Levenshtein distance between it and the most likely word in suggested candidates from the dictionaries is less than

or equal to 1. This setting is designed to filter numerical parts (e.g., 200mm) that are assumed to be correct since spelling errors of such parts are less likely to happen and checking this part may introduce unnecessary complexity. After that, the corrected forms are added to the word container together with the original forms in previous steps. Finally, the words kept in the word container are joined with the separators kept before to generate the corrected results. Note that single words and correct values also work in this algorithm.

3.3 Knowledge Library for Automatic QTO

This step uses the knowledge from the data model and the data from the BIM model to design QTO algorithms for calculating quantities of the targeted elements. As Figure 9 shows, design information (e.g., geometry, spatial relation, material and properties) is extracted from BIM models and applied with identified rule libraries from the standard and data model to determine required variables for the measurement of quantities, rather than taking the geometric data into the calculation directly.

For example, a rule library identified from the HKSMM4 for taking off the quantity of a beam in an in-situ concrete beam-suspended slab joint is shown in equations (1) – (3), V_{beam} and V_{void} are the volumes of the beam and void, respectively, l_{beam} and w_{beam} are the length and width of the beam and slab, respectively, h_{beam} is the calculated height of the beam, H_{beam} is the height from the bottom of the beam to the top of the slab, t_{slab} is the thickness of the slab, C_{beam} and C_{slab} are concrete grades of the beam and slab, respectively. In other words, when the concrete grades of the beam and slab are the same, the slab has higher priority than the beam, and only voids with volumes larger than $0.05 m^3$ are considered in the calculation.

$$V_{beam} = l_{beam} \times w_{beam} \times h_{beam} - V_{void} \quad (1)$$

$$h_{beam} = \begin{cases} H_{beam} - t_{slab}, & C_{beam} = C_{slab} \\ H_{beam}, & C_{beam} \neq C_{slab} \end{cases} \quad (2)$$

$$V_{void} = \begin{cases} 0, & V_{void} \leq 0.05m^3 \\ V_{void}, & otherwise \end{cases} \quad (3)$$

4 Illustrative Examples

Six different models of the same in-situ concrete beam-suspended slab joint are used as the illustrative examples to validate the proposed method through Revit 2021 and Dynamo 2.10 [12]. As shown in Figure 7, they are the same joint created by different modelling approaches, which are not uncommon to see in practice. Besides, Figure 8 shows that the type name is misspelled, and the “concrete grade” parameter is typed as “concete grade”.

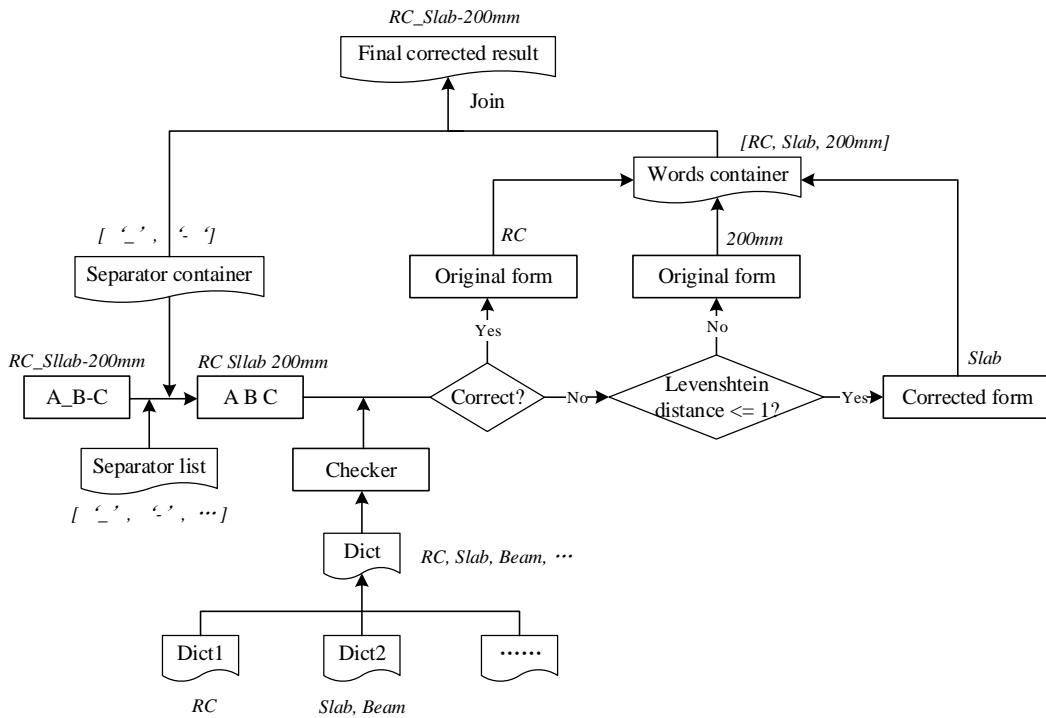


Figure 5. Correctness checking and auto-correction of textual errors

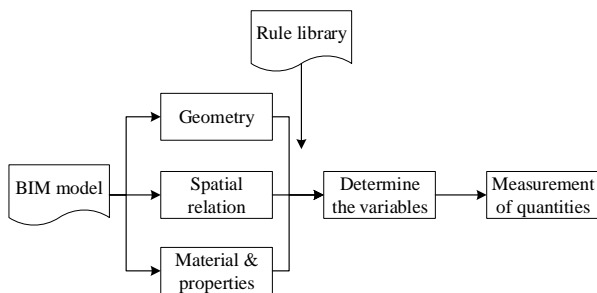


Figure 6. Knowledge-based automatic QTO

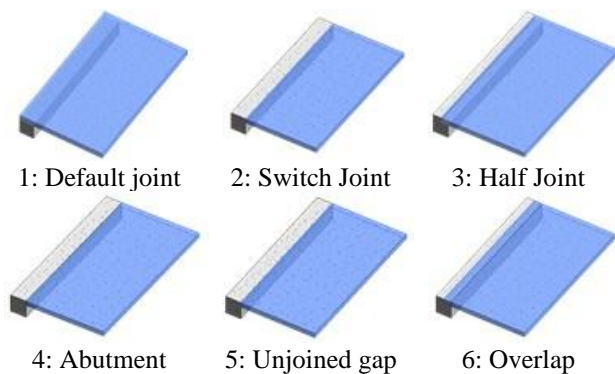
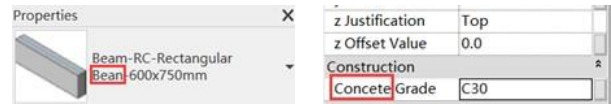


Figure 7. In-situ concrete beam-suspended slab joint models in Revit using different model creation approaches



(a). Misspelled type name (b). Misspelled parameter name

Figure 8. Textural errors of parameters

Based on the generic data model, the information requirements for QTO in this structure are summarized and mapped with the Revit parameters to generate a checklist, as shown in Table 1. Since unit is a project-level parameter and every Revit project has units, there is no need to check the existence of unit parameters for the elements. Of note is that in addition to build-in parameters, users can add project parameters or use customized families in Revit 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”. So, 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. For the illustrative examples, the beam uses the system beam type. The attributes in the right column will serve as the baseline to check whether the information from the model is adequate or not.

Table 1 Mapping required information with Revit attributes for the beam model

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

The information checking algorithms are implemented in Python. PyEnchant is used as the NLP checker to perform the exact and fussy matchings. Results are exported to an Excel file for later uses such as confirmation of checking results and modification of the model. As Figure 9 shows, the beam model contains sufficient information for QTO but there are two textural errors (“bean” and “concrete”) needed to be corrected. It indicates that the algorithms can still recognize the existence of the concrete grade parameter although it is spelled as “Concete Grade” that may be missed in the normal keyword checking method, and the algorithms can recognize and correct the misspelled type name “Bean-600×700mm” and attribute “Concete Grade”.

| | A | B | C |
|---|--------|-----------------------|--------------------|
| 1 | ID | Adequate information? | Missed information |
| 2 | 658757 | Yes | NA |

(a). Misspelled type name

| | A | B | C |
|---|--------|----------------|----------------|
| 1 | ID | Wrong | Correct |
| 2 | 658757 | bean-600×700mm | beam-600×700mm |
| 3 | 658757 | Concete Grade | Concrete Grade |

(b). Misspelled parameter name

Figure 9. Information checking results of the beam

After the information checking step, the QTO system is developed using Dynamo. The Dynamo scripts extract the design information from the Revit model, including geometry and material properties, then unionize the geometries together as a whole. After that, the scripts introduce the concept of cutting plane as the rules. For example, if the concrete grades of the beam and slab are the same, the soffit plane of the slab will be used to cut the unionized solid, otherwise, the inner vertical side of the beam will be the cutting plane. Figure 10 shows how example 1 is cut when the concrete grades of the beam and slab are the same.

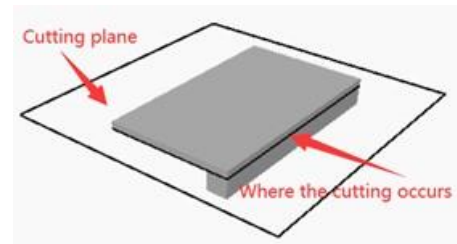


Figure 10. Solid being cut

Table 2 shows the comparison of the quantities of slab using Revit schedule function and the proposed method under the same and different concrete grades (similar comparison results for the beam). The correct quantities are calculated by the authors according to the Hong Kong Standard Method of Measurement of Building Works – Fourth Edition [10] and verified as specification-compliant by an experienced quantity surveyor. The results from the proposed method are exactly correct, while Revit schedule function gives different quantities in some cases. The reason is that the Revit function extracts quantities according to pure 3D geometries without the consideration of the approaches of model creation and semantic data carried by the model, while the proposed method follows the specifications and is able to validate the information and consider both geometrics and semantics.

Table 2. Results of slab’s quantities using Revit schedule and the proposed method

| Case | The same concrete grade (m^3) | | | Different concrete grades (m^3) | | |
|------|-----------------------------------|-----------------|--------------------|-------------------------------------|-----------------|--------------------|
| | Revit schedule | Proposed method | Correct quantities | Revit schedule | Proposed method | Correct quantities |
| 1 | 4.32 | 4.32 | 4.32 | 4.32 | 3.6 | 3.6 |
| 2 | 3.6 | 4.32 | 4.32 | 3.6 | 3.6 | 3.6 |
| 3 | 3.96 | 4.32 | 4.32 | 3.96 | 3.6 | 3.6 |
| 4 | 3.6 | 4.32 | 4.32 | 3.6 | 3.6 | 3.6 |
| 5 | 3.6 | 4.32 | 4.32 | 3.6 | 3.6 | 3.6 |
| 6 | 3.96 | 4.32 | 4.32 | 3.96 | 3.6 | 3.6 |

5 Conclusions

This paper represents a knowledge-based system for more automatic and accurate QTO. This includes the development of a genetic data model based on SMM and domain knowledge, and a textual mapping method for comprehensive BIM model information checking. The SMM descriptive rules are formulated in a knowledge library to support the computation of building quantities. While BIM models can be created by different methods which cause graphical and/or non-graphical errors and impact the calculation of quantities, the proposed knowledge-based system incorporating the SMM descriptive rules can provide more automatic and accurate results.

Six illustrative examples of the in-situ concrete beam-suspended slab joint shows that the proposed method can automate the QTO workflow with little adjustment of the inaccurate BIM models to get accurate quantities in different situations. However, this paper only covers the in-situ concrete for two basic building components. It has room for improvement in respect of the scalability. More comprehensive cases (e.g., a whole structure, unmodelled elements like formworks, scaffolding, and excavations) need to be tested to deal with the complexity of a construction project and large amount of complex rules in the future.

References

- [1] Aram, S., Eastman, C., and Sacks, R. A knowledge-based framework for quantity takeoff and cost estimation in the AEC industry using BIM. In *Proceedings of the 31st International Symposium on Automation and Robotics in Construction and Mining (ISARC)*, pages 434–442, Sydney, Australia, 2014.
- [2] Sacks, R., Eastman, C., Lee, G., and Teicholz, P. *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, 3rd ed.*, John Wiley & Sons, Hoboken, NJ, USA, 2018.
- [3] Sattineni, A., and Bradford, R. H. Estimating with BIM: A Survey of US Construction Companies. In *Proceedings of the 28st International Symposium on Automation and Robotics in Construction and Mining (ISARC)*, pages 564–569, Seoul, Korea, 2011.
- [4] Jadid, M. N., and Idrees, M. M. Cost estimation of structural skeleton using an interactive automation algorithm: A conceptual approach. *Automation in Construction*, 16(6): 797–805, 2007.
- [5] Bečvarovská, R., and Matějka, P. Comparative Analysis of Creating Traditional Quantity Takeoff Method and Using a BIM Tool. In *Construction Maecconomics Conference*, 2014.
- [6] Choi J., Kim H., and Kim I. Open BIM-based quantity takeoff system for schematic estimation of building frame in early design stage. *Journal of Computational Design and Engineering*, 2:16-25, 2015.
- [7] Liu, H., Lu, M., and Al-Hussein, M. Ontology-based semantic approach for construction-oriented quantity take-off from BIM models in the light-frame building industry. *Advanced Engineering Informatics*, 30(2): 190–207, 2016.
- [8] Abanda, F. H., Kamsu-Foguem, B., and Tah, J. H. M. BIM – New rules of measurement ontology for construction cost estimation. *Engineering Science and Technology, an International Journal*, 20(2): 443–459, 2017.
- [9] Staub-French, S., Fischer, M., Kunz, J., and Paulson, B. A generic feature-driven activity-based cost estimation process. *Advanced Engineering Informatics*, 17(1): 23–39, 2003.
- [10] Hong Kong Institute of Surveyors. *Hong Kong Standard Method of Measurement of Building Works, Fourth Edition*, Pace Publishing Limited, Hong Kong, 2005.
- [11] Levenshtein, V. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8): 707–710, 1966.
- [12] Dynamo BIM. Online: <https://dynamobim.org/>, Accessed: 10/01/2021.