Classification and Exemplary BIM Models Development of Design Changes

J.R. Lin[a,*], Y.C. Zhou[b], J.P. Zhang[a], and Z.Z. Hu[a]

[a]Department of Civil Engineering, Tsinghua University, China
[b]College of Civil Engineering, Tongji University, China
E-mail: lin611@tsinghua.edu.cn, zhouyucheng98@qq.com, zhangjp@tsinghua.edu.cn, huzhenzhong@tsinghua.edu.cn

Abstract –

Detection of design changes is essential for collaboration and version management in the design process of buildings. However, current detection methods based on Building Information Modeling (BIM) usually cause unreliable or meaningless results. This is because most of the current researches look at the question from a data change view, which sometimes is meaningless from a designer's view. To overcome this problem, this paper first classifies and identifies meaningful design changes from a designer's view, and develops exemplary BIM models of typical design changes. In this paper, categories of data changes are divided into property data, appearance data and relationship data, and design changes are classified into instance level, type level and model level from a designer's view. The test of two BIM tools (Autodesk BIM360 and IFCdiff) with developed BIM models shows that the detection results for changes at instance level are perfect while detection results for type and model level still need to be further improved. This work contributes a new view and classification method of design changes, and also sets up a baseline model database for further development and validation of relevant methods and tools.

Keywords –

Design Changes; Classification; Building Information Modeling (BIM); Exemplary Model Database; Validation

1 Introduction

The design process of Architecture, Engineering and Construction (AEC) is complex. It includes a lot of iterative work and involves multidisciplinary design work that can be done sequentially, concurrently or in parallel [1] .Changes of a project are inevitable, which can occur from multi-source in any time, and have extensive impact [2]. Accordingly, ensuring the accuracy and consistency of data after sharing and emerging is significant, which raises the requirement of detection of design changes. Building Information Modeling (BIM) is widely used for building lifecycle management. However, current methods for design change detection of BIM models are still not perfect.

Generally, current methods are mainly based on comparing all properties of each instance in BIM models one by one, which is not always meaningful for designers. For example, exchanging the location of two columns that have the same properties should not be detected as a design change in spite of changes of location. To contribute to the development and validation of detection, this paper classifies and identifies meaningful design changes from a designer's view, and develops exemplary BIM models of typical design changes.

This paper is structured as follows. Section 2 reviews the related works and analyses some existing detection methods. Section 3 lists all possible design changes by classifying it by different categories and levels. Section 4 uses exemplary BIM models developed to test some BIM tools and summarizes their results. Finally, section 5 concludes this paper's contribution.

2 Literature Review

Design changes in the BIM model has been extensively studied by researchers. These research works included the management of design changes by versioning Industry Foundation Classes (IFC) models [3] or IFC extension [4], improving design coordination [5], quantifying the design change time ripple effect [6], and assessing impacts of construction change orders [7]. Even though these researches made design changes to be managed and evaluated more effectively, few works focused on the definition and classification of meaningful design changes from a designer's view.

The most common method used for detecting design changes between two BIM models is based on the ID (e.g. GlobalId in IFC file or ElementId in Autodesk Revit) and comparing all properties between two instances, which is adopted in the above references and most commercial BIM platforms such as Autodesk Revit, Navisworks, etc..

However, ID is unstable sometimes, which will lead to wrong and unnecessary detection results. Meanwhile, its detection results are not always meaningful. For example, exchanging the location of two same columns will be detected as a change by this method, but that is a meaningless change for designers actually.

Daum, S. and A. Borrmann [8] proposed an approach for detecting equivalences in datasets of the Industry Foundation Classes (IFC), which based on a geometrical matching and semantic comparison method. This approach is independent of ID, which means the detection can be completed even IDs are not reliable. Nevertheless, meaningless results will still be detected by this method.

Shi, X., et al. [9] proposed a content-based automatic comparison approach for IFC files. This approach will construct a hierarchical structure for both files and then use an iterative bottom-up procedure to compare them. However, the test results of detecting changes in some cases such as exchanging location of two same columns and deleting and recreating a same beam are still not perfect.

In summary, most of the current methods only look at the problem from a data change view, while ignore the designers' view. To overcome this problem, the design changes should be carefully reviewed and classified.

3 Classification of Design Changes

As an open BIM model data representation standard, IFC defines an express based entity-relationship model consisting of several hundred entities organized into an object-based inheritance hierarchy [10]. And in Autodesk Revit, the data in a document consists primarily of a collection of elements [11], where each element has many properties. It can be seen that BIM models can be described by instances (objects), which consist of many properties.

Design change detection is usually between an old and a new files (documents). Previously, design changes can be divided into 3 change types from a data change view as follows.

- Added. An instance is created in the new file.
- Deleted. An instance is deleted in the old file.
- Modified. An instance exists in both old and new file but its properties has been changed.

However, it is not enough to simply classify design changes as the above three types. Because that may lead to meaningless results and the results can be further optimized. To address this issue while ensuring all the changes can be properly identified from a data change view, this paper firstly divides data changes of BIM into three categories, namely, property data, appearance data and relationship data. Then, all possible design changes are classified into three levels, instance level, type level, and model level respectively from a designer's view.

3.1 Categories of Data Changes

The data changes of BIM can be divided into three categories as follows.

- Property data: represents instances' properties, such as parameters or other user specified attributes.
- Appearance data: represents instances' 3D appearance, such as the geometry and location.
- Relationship data: represents the relationship between two instances, such as the relationship that a wall hosts a door.

As for property data, the possible changes are:

1. Added. Add a new property of an instance.
2. Deleted. Delete an existing property of an instance.
3. Value Modified. Modify a property's value.
4. Name Modified. Modify a property's name.
5. Order Changed. Change the order of some properties of an instance.

As for appearance data, the possible changes are:

1. Added. Add a new geometry of an instance.
2. Deleted. Delete an existing geometry of an instance.
3. Geometry Modified. Modify a geometric shape of an instance.
4. Transformation Modified. Modify a transformation property of an instance. Such as scaling, translation and rotation.
5. Representation Method Modified. Change the way of representation of a geometry. Such as converting a solid model to a surface model.

As for the relationship data (assuming that the order of relationships is meaningless), the possible changes are:

1. Added. Add a new relationship between two instances.
2. Deleted. Delete an existing relationship between two instances.
3. Instance Modified. Modify property data of a relationship instance.
4. Relationship Modified. Modify a relationship between two instances, e.g. change the relationship between A and B to A and C.

3.2 Levels of Design Changes

The possible data changes given above should be identified as least at the instance level. Thus, adding and deleting properties of an instance should be identified as changing an instance, so as modifying some properties of an instance (Left part of Figure 1). However, for meaningful and optimal identification results, the

perspectives of type level and model level are required.

From the perspective of type level, the changing of instances may be caused by changing of a type (or family type in Revit) of the instances. For example, the geometric changes of multiple instances caused by changing the geometry of the type that all the instances inherit from should be correctly detected (Middle part of Figure 1), which can reduce the detection results and make the results more meaningful.

From the perspective of model level, data changes or modification of instances may be meaningless sometimes.

For example, exchanging the location of two same columns should not be identified as a change, so as deleting a beam and recreating a same one (Right part of Figure 1). Besides, modification of the elevation of a Level may result in a large number of changes in related instances. So identifying the source of the change is better than just detecting the changed related instances, and that will contribute to more meaningful results. Changes from the perspective of these three levels are illustrated in Figure 1.
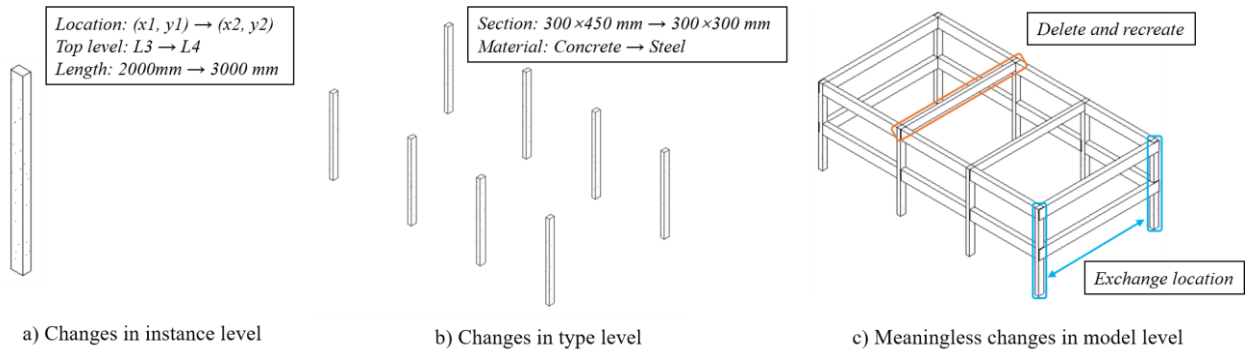

Location: (x1, y1) → (x2, y2)
Top level: L3 → L4
Length: 2000mm → 3000 mm

Section: 300×450 mm → 300×300 mm
Material: Concrete → Steel

Delete and recreate

Exchange location

a) Changes in instance level

b) Changes in type level

c) Meaningless changes in model level

Figure 1. Design changes from perspective of instance level, type level and model level

4 Exemplary Models and Case Study

4.1 Exemplary BIM Models

Since there is no public available database for testing and verifying current design change detection methods, this research develops 11 simple BIM models according to the changes described in section 3. In these changes, added or deleted in all types of data are combined to facilitate test. And the order change of property data and representation method change of appearance data are ignored in this paper. Meanwhile, type level design changes and meaningless design changes at model level are also embedded in the developed BIM models. Detailed information of these models are shown in Table 1, and their screenshots are shown in Figure 2.

Table 1. The detailed information of exemplary BIM models

Model Name	Category of Data change [a]	Level of Design Change	Description of Changes	Meaningful [b]
M1	-	-	The origin model	-
M1_All-A	Added in P, A and R	Instance	Add 2 beams and 2 columns (new category)	○
M1_All-D	Deleted in P, A and R	Instance	Delete 2 beams and 4 columns	○
M1_All-DA(M)	Deleted and then added in P, A and R	Model	Delete 2 beams and 4 columns and recreate them	✗
M1_A-MG	Geometry modified in A	Instance	Modify the sectional dimensions of 2 beams	○
M1_A-MG(T)	Geometry modified in A	Type	Modify the section of family of a beam from 400×800 to 800×400	○
M1_A-ML	Location modified in A	Instance	Modify the location of 4 columns and 2 beams	○
M1_A-ML(M)	Location modified in A	Model	Exchange the location of 2 pairs of columns and 1 pair of beams	✗
M1_P-MV	Value modified in P	Instance	Modify a parameter (COMMENTS) in a beam	○
M1_R-MI(M)	Instance modified in R	Model	Modify the elevation of a level (Level3)	○
M1_R-MR	Relationship modified in R	Instance	Modify the top level of 4 columns	○

[a] P, A, R means property data, appearance data, relationship data respectively.
[b] ○ and ✗ means the change is meaningful and meaningless respectively.
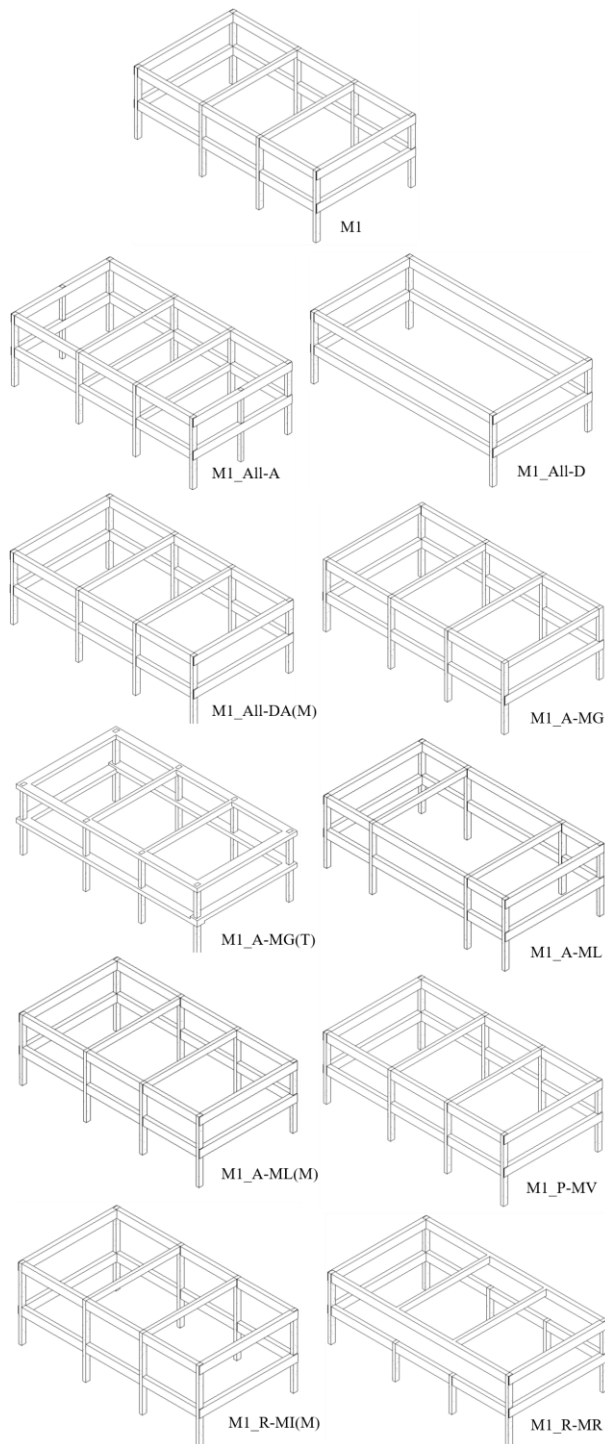
Figure 2. Exemplary BIM models

It is apparently that the model named M1 has 8 columns and 10 beams, and the changes from M1 to any other model can be used as a test case for design change detection methods. A repository contains these exemplary BIM models and their detailed description are

established on GitHub. Each model has a .rvt and .ifc file (the contents of the files in these two different formats are the same). The link address is: https://github.com/smartaec/Design-Change-BIM-Models.

4.2 Test of Related Systems

This paper selects Autodesk BIM360 and the IFCdiff [9] to verify their accuracy of detection (according to the file format they support, BIM360 uses the .rvt files while the IFCdiff uses the corresponding .ifc files).

The ideal test results are all changes at instance level are detected. Meanwhile, for changes at type level, the source of changes should be detected instead of the direct results. And for changes at model level, the results should be that nothing has changed because changes in these cases are meaningless.

4.2.1 Test Result of Autodesk BIM360

BIM360 has the correct results for all test cases of changes at instance level, but its results for type and model level are incorrect. The screenshots of detection results in M1_A-MG(T) and M1_All-DA(M) using BIM360 are shown in Figure 3.
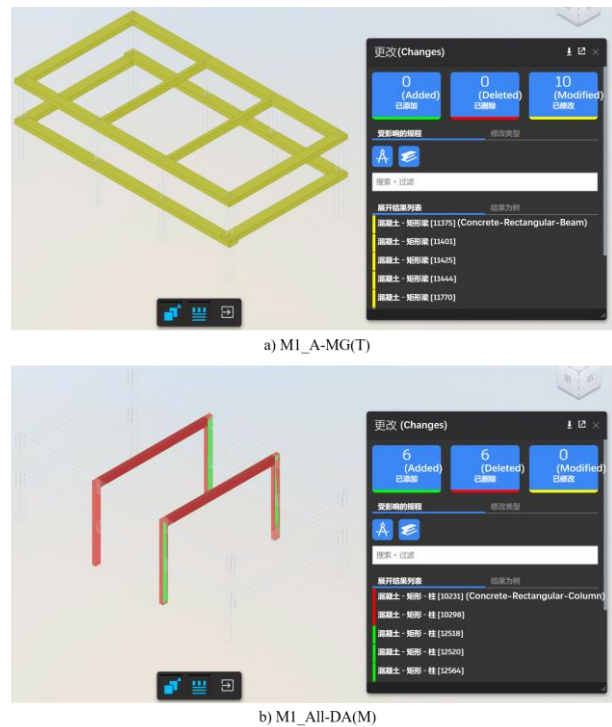

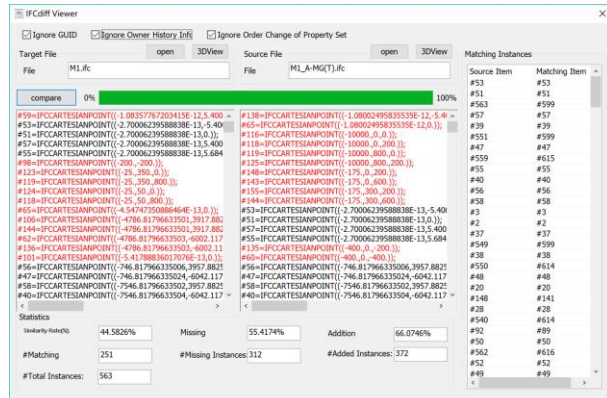Figure 3. The detection result of M1_A-MG(T) and M1_All-DA(M) by BIM360.

In Figure 3 a), 10 modified changes are detected. 8 beams of them are changed in section size while other 2

beams are changed in volume because of the affect in geometry of joint. Actually, the only change is the change of the section dimension of a family of beam from 400×800 to 800×400 mm, which is a change at type level. However, BIM360 detected the changes of the affected 8 beams but doesn't identified that the family of these beams has changed. So, BIM360 is not as good as we expected.

In Figure 3 b), 6 added and 6 deleted changes are detected. The actual operations in this model are delete 6 components and recreate 6 same one. And the ground truth from a designer's view is that nothing has changed since these changes are meaningless. So, BIM360 cannot handle the model level changes properly now.

4.2.2 Test Result of IFCdiff

Similar to BIM360, IFCdiff has correct results for all test cases of changes at instance level and incorrect results for type level. But for changes at model level, its results are better than BIM360. The screenshots of detection results in M1_A-MG(T) and M1_All-DA(M) using IFCdiff are shown in Figure 4 (the output of IFCdiff is text).
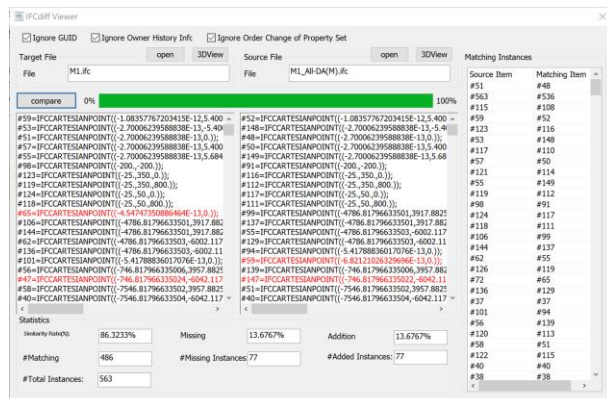
In IFCdiff, the similarity rate (%) is defined as the rate of the number of identical instances between File A (the target file) and File B (the source file) divided by the total number of instances in File A (both files have been removed redundant instances), i.e.

$$\text{Similarity}(A, B)(\%) = \frac{|A \cap B|}{|A|}.$$

In Figure 4 a), File A (i.e. M1) has 563 instances and File B (i.e. M1_A-MG(T)) has 623 (251+372) instances. The number of matching, missing and addition are 251, 312 and 372 respectively. From the result of BIM360, 10 of 18 components have been changed, about 44.44% (8/18) of the components have not changed. So, the similarity rate 44.58% given by IFCdiff is reasonable. But it also cannot identified that the family of these beams has changed from the perspective of the type level. Therefore, IFCdiff cannot handle type level changes properly.

In Figure 4 b), the similarity rate is 86.3% and the added and deleted rates are both 13.6%. In contrast, the test result of BIM360 is 6 added and 6 deleted. It can be inferred that the added or deleted rate is about 33.3% (6/18) detected by BIM360. Thus, the result of IFCdiff is a bit better than BIM360 but still not perfect.

In the authors' opinion, the reason why test result of IFCdiff is better than BIM360 in the latter case is IFCdiff can ignore changes of GlobalId, Owner History, etc. when comparing two instances. In the latter case, the deleting and recreating same instances are equivalent to updating their ID, so IFCdiff will detect less changes. But for the former test case, there is no essential difference between IFCdiff and BIM360 when detecting changes at type level, so neither can give a perfect result.

4.3 Summary of Test Results

According to the results of the case study and analysis, Table 2 summarizes the correctness of the two tools.

a) M1_A-MG(T)

b) M1_All-DA(M)

Figure 4. The detection result of M1_A-MG(T) and M1_All-DA(M) by IFCdiff.

Table 2. The detection results of BIM360 and IFCdiff using developed exemplary models

Model Name	BIM360	IFCdiff
M1_All-A	O	O
M1_All-D	O	O
M1_All-DA(M)	X	*
M1_A-MG	O	O
M1_A-MG(T)	X	X
M1_A-ML	O	O
M1_A-ML(M)	X	*
M1_P-MV	O	O
M1_R-MI(M)	X	X
M1_R-MR	O	O

O means the result is correct (only this means passed the test).

X means the result is not completely correct.

＊ means the result is not completely correct but better than X.

It can be seen from the Table 2 that both BIM360 and IFCdiff can passed tests when detecting changes at instance level but cannot deal with type and model level changes now. Meanwhile, the detection results of IFCdiff for changes at model level are better than BIM360, since IFCdiff will get fewer changed results when detecting meaningless changes.

5 Conclusion

This paper proposes a new view and classification method of design changes, and also sets up a baseline model database for further development and validation of relevant methods and tools. Three categories of data changes (property data, appearance data and relationship data) and 3 levels of design changes (instance level, type level and model level) are considered, and corresponding exemplary BIM models are developed. Then, this paper selects Autodesk BIM360 and IFCdiff to verify their accuracy of design change detection. The results show that both BIM360 and IFCdiff can detect changes at instance level well but cannot handle type and model level changes properly.

This work reveals current problems in design change detection and introduces new method on classification of design changes. Meanwhile, this paper first sets up an exemplary BIM model database of design changes, which can be used as baseline data for the validation of future methods and tools for design change detection.

Acknowledgements

References

[1] Nour, M. and K. Beucke. Object versioning as a basis for design change management within a BIM context. In *International Conference on Computing in Civil and Building Engineering (ICCCBE-XIII)*, pages 147–152, Nottingham, UK, 2010.

[2] Motawa, I.A., et al. An integrated system for change management in construction. *Automation in Construction*, 16(3): 368–377, 2007.

[3] Jaly-Zada, A. and W. Tizani, Design Change Management Based on Versioning the IFC Models. In *16th International Conference on Computing in Civil and Building Engineering*, 520–526, Osaka, Japan, 2016.

[4] Jaly-Zada, A., C. Koch and W. Tizani. IFC Extension for Design Change Management. In *32nd CIB W78 Conference 2015*, 328-335, Eindhoven, Netherlands 2015.

[5] Kim, H. and F. Grobler, Design Coordination in Building Information Modeling (BIM) Using Ontological Consistency Checking. *Computing in Civil Engineering (2009)*, 410–420. ASCE, https://doi.org/10.1061/41052(346)41, 2009.

[6] Moayeri, V., O. Moselhi and Z. Zhu, BIM-based model for quantifying the design change time ripple effect. *Canadian Journal of Civil Engineering*, 44(8): 626–642, 2017.

[7] Likhitruangsilp, V., T.N. Handayani and N. Yabuki, A BIM-Enabled Change Detection System for Assessing Impacts of Construction Change Orders. In *17th International Conference on Computing in Civil and Building Engineering*, Tampere, Finland 2018.

[8] Daum, S. and A. Borrmann. Enhanced Differencing and Merging of IFC Data by Processing Spatial, Semantic and Relational Model Aspects. In *23rd International Workshop of the European Group for Intelligent Computing in Engineering*, Kraków, Poland, 2016.

[9] Shi, X., et al., IFCdiff: A content-based automatic comparison approach for IFC files. *Automation in Construction*, 86: 53–68, 2018

[10] Wikipedia. Industry Foundation Classes. Online: https://en.wikipedia.org/wiki/Industry_Foundation_Classes, Accessed: 12/12/2018.

[11] Autodesk, Element Class. On-line: http://www.revitapidocs.com/2018.1/eb16114f-69ea-f4de-0d0d-f7388b105a16.htm, Accessed: 12/12/2018.