IFC EXTENSION FOR DESIGN CHANGE MANAGEMENT

Aras Jaly-Zada, evxaj2@nottingham.ac.uk
The University of Nottingham, United Kingdom

Christian Koch, christian.koch@nottingham.ac.uk
The University of Nottingham, United Kingdom

Walid Tizani, walid.tizani@nottingham.ac.uk
The University of Nottingham, United Kingdom

Abstract
Building design is a comprehensive, iterative, and distributive process. It generates large amounts of data that require transfer among multi-disciplinary designers at various design stages. Building Information Modelling (BIM), as a method to support the entire building life-cycle, including the design, and the Industry Foundation Classes (IFC), as an open data representation of the current building state within the BIM process, play an important role in the process of collaborative design. The mechanism to deal with changes to the building information among different models needs more studies to keep pace with the evolution in BIM and IFC. Making changes in the properties of the building elements by any designer need to be done with the consent of others to ensure consistency. The current IFC standard does not take into account the sharing of changes among different BIM models nor the recording of the history of earlier changes. In this paper, we propose an extension to the capability of the existing IFC EXPRESS schema in order to deal with changes in different BIM models from conception to completion. We also present a newly developed structured versioning system to enable effective collaboration. The versioning mechanism is integrated within the actual design model and allows recording of changes in the model as a new version within the IFC model. A prototype system is implemented using the .NET framework and linked into Revit. This system demonstrates that managing changes through the extended IFC schema can improve collaborative design.

Keywords: BIM, IFC extension, versioning, collaboration

1 Introduction
The building design process includes a series of activities undertaken from the inception of the project to the handing over to provide a complete set of drawings for the construction process. Different levels of expertise participate in the process and usually work on different BIM models depending on their specialization. Those BIM models are large, complex, and interdependent. There are some pieces of information in BIM shared among different designers that can be used several times by them. Beams, columns, walls, etc.... are shared elements among all designers and need to be used within the design package for each of them while, steel bars, cooling ducts, furniture, etc… are specific elements of one type of disciplinary designers. Changes in design are inevitable even with the contemporary of BIM. The contents of the design changes in each one of the BIM models must be clarified and transferred clearly to the others to make sure that all are working on the latest revision of the model (Macdonald, 2013). Managing those changes is a core concern of any design process. Model versioning is a way to support change management that can be implemented at the

very early design stage to improve the collaborative design. Thus, the design has to pass through several versions to represent different development stages and different design solutions. Versioning has long been in the focus of scientific researchers for various purposes. Nour and Beucke (2010) clarified that since building elements are represented in the form of objects, information management systems could be proposed at the object level, through versioning models and managing changes in the objects. Koch and Firmenich (2011) proposed a new processing-oriented building model through integrating the existed state-oriented descriptions with the additional change-oriented information, as modelling operations. Based on the version-oriented method, the change–oriented approach is introduced. Jaly-Zada et al. (2014) proposed a collaborative versioning system among multi-disciplinary designers to track the affected changes in the data model.

As in many other industrial sectors, a major difficulty that the engineering sector is currently facing is the lack of BIM interoperability between software applications (Karan and Irizarry, 2015). One application does not necessarily allow reading the BIM models from the others owing to incompatibilities among them. It generates a high level of re-working on information and leads to a lack of integration between the different stages in the design processes and hamper collaborative design (Oti and Tizani, 2014). Many data exchange standards are already in use and under development. IFC is the most powerful data model to facilitate interoperability between AEC software applications. It is developed by buildingSMART (formerly the International Alliance for Interoperability, IAI). IFC defines an EXPRESS based entity-relationship model containing several hundred entities structured into an object-based inheritance hierarchy. The existing IFC EXPRESS schema deals with the design changes at a very simple level. An enumeration type *IfcChangeActionEnum* within the IFC schema defines the actions associated with the recent changes made to the objects (such as, 'added', 'modified') without defining the value of the changes. *IfcOwnerHistory* is the only entity that references *IfcChangeActionEnum*. This means that it is used with the entities stemming from *IfcRoot* (such as, beams, columns…), but not with the entities belonging to the resources data schemas. Therefore, the current mechanism in IFC cannot track the changed information in the features of the building elements, so that the instance that represent the profile section of the beam, for example, cannot be considered as changed information in the IFC standard.

Researchers have suggested extensions to the IFC schemas in different domains to better facilitate interoperability. Kléos et al. (2012) defined new attributes to the existing entity "*ifcStructuralLoadGroup*", this entity is responsible for defining load groups, load cases, and different combinations. The new attributes allow the load safety factors used by many codes of practices to be defined to hold the maximum and the minimum design load from different load cases. Gökçe et al. (2013) presented an extension with new entities and new relations in the frame of the "*IfcConstructionResource*" entity complying with the data schema of the IFC standard for the management of cost based on material analysis information. Some open source platforms have been designed and implemented by some researchers. For instance, the BIMserver software is for supporting an early warning system (Beetz et al., 2010). It informs users to avoid inconsistency of models between different users. This system is a server based but not an engineering design based. Tekla Corporation and Solibri Model Checker (SMC) have developed BCF (BIM Collaboration Format) to enable workflow communication between some BIM software tools. The BCF idea is to encode messages containing BIM-topics (e.g. issues, proposals, change requests…) addressed in BIM-data-models using IFC mechanisms for Global Unique ID's (GUIDs). BCF format has been built into limited software packages and it is more designed to improve the communication among designers rather to versioning the changes in BIM models.

The main goal of this study is to integrate an object-oriented data model, as a representation of the current information in BIM, and the change management, as a representation of the changing information in BIM to introduce an extended data model for design change management. The new data model covers the changed and unchanged information to provide a foundation for managing changes in BIM and collaborating multi-disciplinary designers. An extension to the capability of the existing IFC EXPRESS schema is suggested to deal with the different changes in different BIM models from conception to completion. New entities are provided as new data structure, which were not specified in the current IFC standard, to enable dealing with the changes. The history of

involvement of building elements can be embedded within the neutral data file so that it becomes possible to extract, keep track, retrieve, and manage changes in different BIM models.
The next section illustrates the changed information in BIM and the versioning concept to manage the new, modified and deleted information. The third section adds the versioning concept into the IFC schema to have multiple versions of the changed information. The implementation of an experimental prototype is presented in section four to test and verify the entire approach. The last section concludes this paper.

2 Versioning the changed information

In order to enhance the collaboration of multi-disciplinary designers to support model changes, a versioning concept is used as an information management system. A new design version represents the state of development at a particular time. Any building element in the BIM and any feature for each element can be defined and managed. The new version of the model can be generated by identifying at least a single difference in comparison with the predecessor-shared version. The differences cover more than geometrical parts of the model element. It includes material, quantities, specifications, calculations, analytical results, and may have some extra information about costs and sustainability for different professions (architecture, structural, mechanical, electrical, and sanitation designers, etc…).
The conventional BIM model does not allow identifying the information that has been changed since the last state. Figure 1 illustrates the current state model that might contain a set of unchanged and changed information. Each circle in figure 1 represents one set of information based on elements states. There are three different types of changes that might have occurred to the building elements since the last model version (add, modify and delete). The deleted elements are missing information in the current model. The proposed information model for versioning includes information about the deleted elements together with the new and modified elements to make up all the changes set that requires being included within the current model.

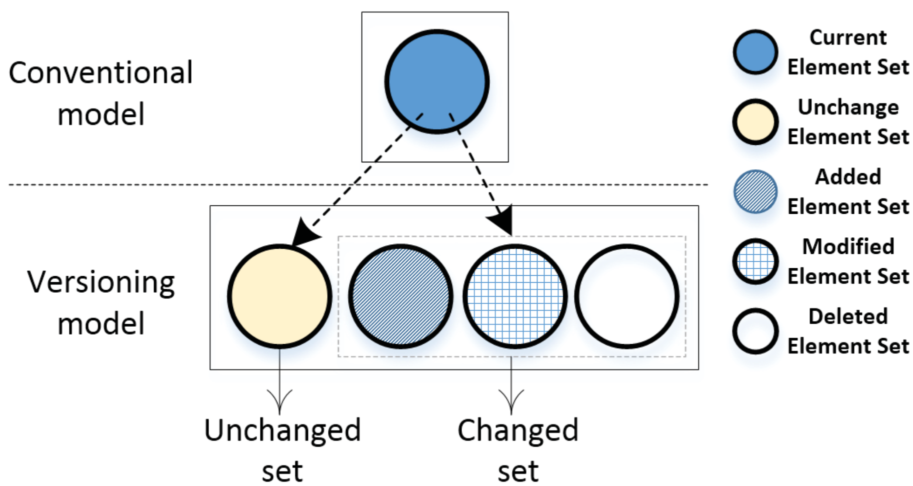

Figure 1: Conventional and versioning models

The information about each element in the same BIM model differs from others in terms of the features information (location, profile shape, material, etc…) that represents the element. Each element in the changed set described above can be classified more precisely to a subset of information that contains the element features. The features for a single element are represented in Figure 2. A circle shape represents an element, while a diamond shape refers to a feature. Available BIM tools represent the current situation of the element, as in Figure 2/a, without identifying the new or modified features or including the deleted features or the old information about the modified features. The features information needs to be easily identified by the designers, so they can receive full information on the changes in the new model version. The ideal clarification of the features information within an element is shown in Figure 2/b with identifying all new, deleted, modified, and unrevised features information.

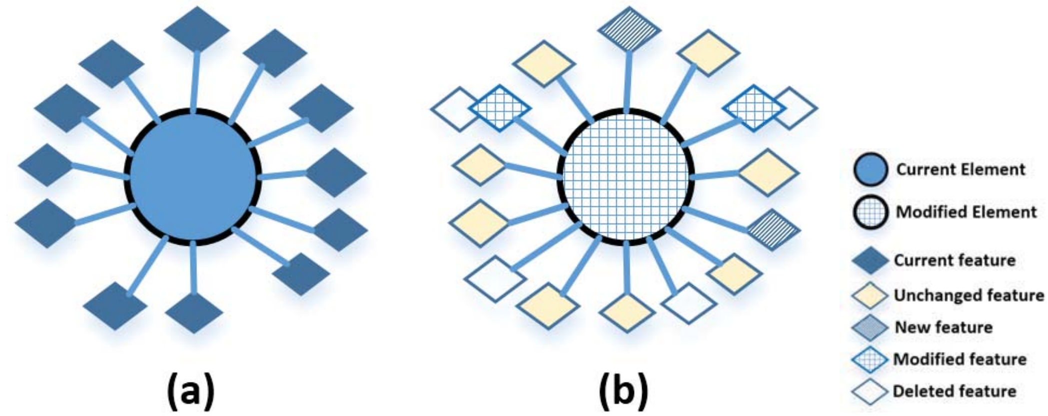

Figure 2: Representation of the features for a single element.

Taking the model as a set of elements with associated features, a simplified example for a model (M) with two elements (E_1, E_2) and two features information for each element (F_1, F_2 for E_1 and F_3, F_4 for E_2) is shown in Figure 3. The figure displays possible developments of several Versions of the Model (M_{vn}) for different Element Versions ($E_{n,vn}$) and Feature Versions ($F_{n,vn}$). Where indices 'n' is the element or feature identifier and 'vn' is the version number respectively. Any version represents the current state at this stage of development. It can be noticed from Figure 3 that the version indices for the model can be different from those of individual elements and features. In M_3, for example, the version of E_1 is $E_{1,2}$ and the version of F_1 is $F_{1,1}$. Thereby, the same feature version can be in several element versions and the same element version can be in a number of different model versions. In Figure 3, change development for the elements and their features in each model version are represented to structure the development in changed information only without the redundant information and to indicate how the various versions are related to each other to identify the changed information. Different versions of the changed model are displayed (C_{vn}) whereas index 'vn' is the same for the changed model (C) and the whole model (M), since (C) is the changed part of (M) and generating a new model version (M) requires at least a single change in the information.

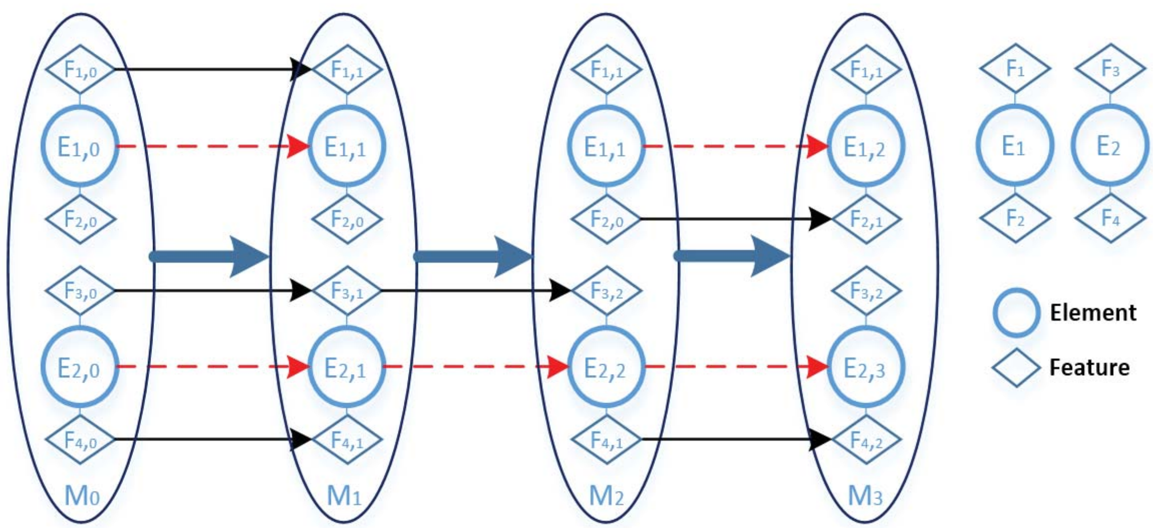

Figure 3: Model development graph.

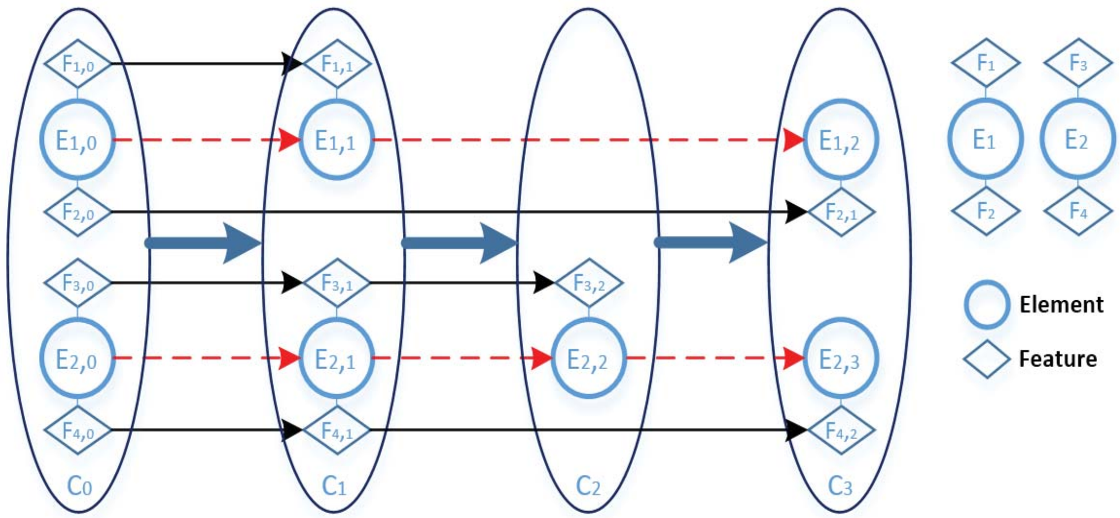

Figure 4: Change development graph.

3 Versioning the IFC model

The IFC standard can be used as a schema for data transfer between various BIM applications. Information models do not deal with the changes in the building elements; this reflected in the data models. Currently, objects in IFC present a last state at this stage of the BIM model; they may contain revised objects over the course of a new version of the model. Furthermore, the IFC standard does not include the human requirements to deal with the information in the design, construction, and facilities management, such as the designers' demand to clarify the changing information in the model. As a designer, It is not required to know what IFC STEP lines have been changed, while the intention is to find out the meaning of the changes in the IFC, in terms of the changes in the elements and features, to represent the human perception. It is not necessary to know that the (*IfcIShapeProfileDef*) object in the IFC file has been changed whereas it is worth to know that a specific section value for a particular element has been changed.

The IFC EXPRESS schema includes inheritance hierarchy of hundreds of entities that represent the building elements, the features of the elements and the relationships among them. There are many complex entities trees to define the function of each entity (buildingSMART, 2015). IfcProduct, for example, is the base entity for all building elements within the IFC EXPRESS definition (IfcBeam, IfcColumn, etc.…). It has attributes that are inherited by all entities of the building elements to define some of the feature information, such as a shape representation (*IfcProductRepresentation*) and an object placement (*IfcObjectPlacement*). The other features definitions (length, cost, colour, etc…) can be linked with the element through an objectified relationship (*IfcRelDefinesByProperties*) to describe the association between a set of features and an object. Figure 5 illustrates the relation of the entities in part of the IFC STEP file to define a specific building element, which is represented by "IfcBeam".Each feature has a map of a complicated combination of entities to represent the feature. Therefore, Information about engineering features (such as, locations, cross-section, material, and quantity) to the building elements can be defined within the EXPRESS definition in references to a set of relation's object entities. These entities are (*IfcCartesianPoint*, *IfcIShapeProfileDef*, *IfcMaterial*, and *IfcPropertySingleValue*) to reflect sequentially the features information above. On the other hands, Information about the latest model version regarding the name and the organization that created the current BIM model, the software application that has been used, the creation date and time to the current IFC model as well as capture the last modified creating and user are all defined in *IfcOwnerHistory* entity. It is an attribute to the root entity. Therefore, it is directly attached to all objects, relationships, and properties entities.

The entities that represent the features are not directly connected to the entity that represents the specific element within the IFC standard. For example, *IfcBeam* is an object instance in IFC STEP file to represent one of the building beams and *IfcIShapeProfileDef* is another object instance to represent one of the features of that beam, which is the profile I-shaped. There is no unique mapping for defining each element feature. Thereby, The IFC schema provides different ways for defining many types of features and linking them with the building element. For example, four associated entities designate the *IfcIShapeProfileDef*. Each entity represents a new object line in the IFC STEP file. For the example in Figure 5, a series of entities required to create the I-shape can be defined as follows:

- *IfcProductDefinitionShape* is a general container for all representations for a product. It allows for a characterization of the product representation by a name and for a provision of further description information
- *IfcShapeRepresentation* is more specific container for product's representation. It carries additional classifications to indicate the representation kind as a "Body" and the representation type as a "solid model".
- *IfcExtrudedAreaSolid* is to generate the swept solid shape of the element. Swept solid is one of the ways to form solid shapes within the IFC schema. The length and the direction of the beam are allocated within this object.
- *IfcIShapeProfileDef* is to allocate the I-section profile of the beam. This section profile is used by the swept solid to form the 3D solid shape. The value of the I-section with its overall depth, width and its web and flange thickness are given within this object.

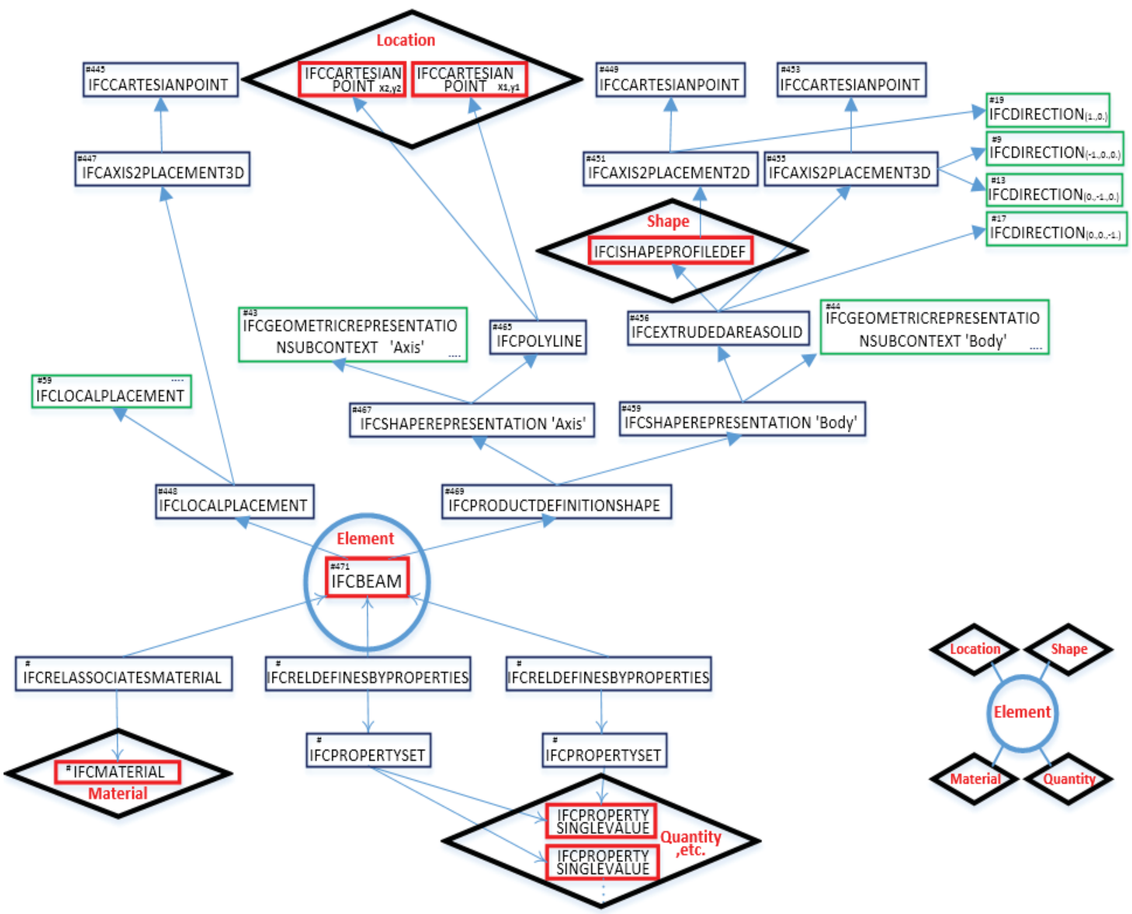

Figure 5: The graphical modelling of a beam in the IFC STEP file.

A series of changes may occur in each type of element features since the release of the latest model version. To manage the new model version, the changed elements and features are needed to be identified and attached to the IFC STEP file. The entities, which are within the specification of the IFC EXPRESS to define the building elements and the features, have been used to define the changed elements and the changed features respectively. Therefore, these entities will be used several times in the IFC STEP file to define the previous feature values of each changed element. As mentioned in the last section, there are three different types of changes (add, modify and delete) since the last model version. Addition and deletion are dealing more with the element because all the element features within the current model version have existed for the added element or removed for the deleted element. Whereas, modification is more dealing with some of the element features. Therefore, new relationship entities (*IfcRelElementChange* and *IfcRelFeatureChange*) have been established to deal with the versioning concept that is not within the EXPRESS schema specification. These entities link the object instances of the changed elements and the changed features respectively with the object instances that are generated to define the general Information about a specific model version (*IfcOwnerHistory*). Therefore, each changed element with their features is connected to the owner history of that element.

Figure 6, which is an extension for a part of figure 5, demonstrates the changes for a specific beam element (ID 170031) and one of their features (I-Profile Section). Two versions have been proposed of the model to represent the changes, the beam element in version one is added and the profile shape in version two is modified. In version one, *IfcRelElementChange* has been added to the IFC STEP file to link the new beam (ID 170031) with the information about the owner of the first version. *IFcBeam* connected indirectly with all the entities that represent the features of that beam. Thereby, the features of the beam are marked implicitly as new features in the first version. In version two, *IfcRelFeatureChange* has been added to the IFC STEP file to link, on one hand, the modified values (the new with the old values of the feature) and to link, on the other hand, the changed feature with the information about the owner of the second version. Within these entities, the versioning concept has been added into the IFC schema. It becomes possible to have multiple versions of each feature of its element. Any change in the element represents a change in the overall BIM model. A version history of the elements can be provided within the model to represent the development of design processes and to demonstrate the sequence of versioning process in terms of adding new information and modifying or deleting the existing information. With the version history, it is possible to navigate through the details of the elements from each version and acquire complete information about the features of each element.

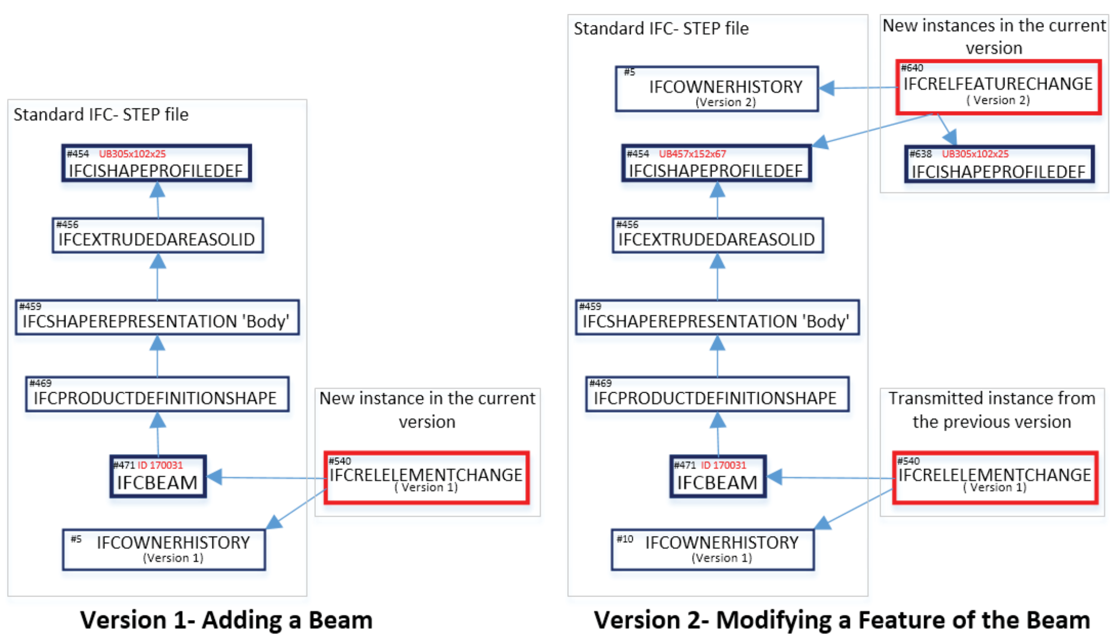

Figure 6: The entities in two versions in the IFC STEP file.

4 Prototype Implementation

In order to verify the recommended approaches, a prototype program is implemented in a .NET framework environment using C# Programming language. Since Revit API allows to program in the .NET compliant language. The prototype program is integrated into Autodesk Revit to gain access to parameter and graphical data of the BIM model, to identify and manage the changes in elements, to automate repetitive tasks, to create an extended IFC and to link multi-disciplinary designers together.

The testing and verifying for the entire approach was done through implementing the prototype program in a typical design activity with the different disciplinary designers' involvement. The BIM project used for the case study is an office building. A set of columns, beams, etc… displayed as changed information within the proposed model. The goal is to show the efficiency of the proposed collaborative modelling framework in supporting the designers to manage changes based on the suggested extension to the IFC model. This IFC model complies with the format of the EXPRESS data modelling. The process begins by calling up the prototype program from Autodesk Revit plug-in where this program has been integrated through Revit Platform API. The program captures relevant changes in the BIM model, adds this information as new IFC object entities, generates a new version within the current IFC model, and manages all previous versions. Figure 4 shows a snapshot of the extended IFC model on the BIM model. The prototype program can present all changes numerically and graphically through drawing the new elements, deleting the existing elements, and modifying the properties of the existing elements in the BIM model. Furthermore, the history of changes to any element of the model on earlier dates can be represented to illustrate all preceding changes.

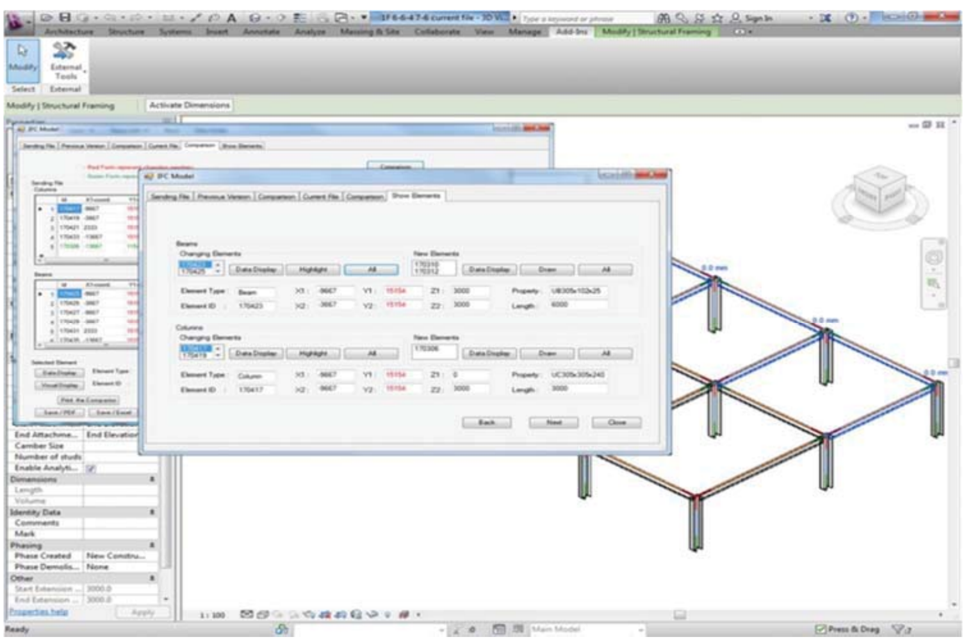

Figure 4: snapshot of the extended IFC model

5 Conclusion

Building design includes a wide range of issues that attract multiple designers from different domains. The collaboration among different BIM users irrevocably is becoming an essential demand. Changes in different models are a fact at all stages of design and construction. Making changes by any designer need to be done with the collaboration of others to ensure consistency. The current IFC standard does not take into account the sharing of changes among different BIM models nor the recording of the history of earlier changes.

This paper has covered some features of design change management that is implemented within the BIM model at the very early design stage. The proposed approach suggests an extension to the capability of the existing IFC EXPRESS schema to deal with different changes in different BIM models from conception to completion. Different entities of the IFC data model are provided as new data structure, which were not specified in the current IFC standard. The suggested approach supports versioning in file exchange and visualizing all changes in the information model. It can be concluded that this research involves processes of storing, managing, versioning, exchanging and sharing of building information in an interoperable and reusable way.

Acknowledgements

This work is sponsored by the KRG-HCDP Scholarship program / Ministry of Higher Education and Scientific Research/ Kurdistan Regional Government.

References

BEETZ, J., VAN BERLO, L., DE LAAT, R. & VAN DEN HELM, P. BIMserver. org–An open source IFC model server. Proceedings of the CIP W78 conference, 2010.

BUILDINGSMART. 2015. *Industry Foundation Classes/ Design Transfer and Reference Model Views* [Online]. http://www.buildingsmart-tech.org/mvd/review/IFC4Add1/DTVRV/candidate/html/.

GÖKÇE, K. U., GÖKÇE, H. U. & KATRANUSCHKOV, P. 2013. IFC-Based Product Catalog Formalization for Software Interoperability in the Construction Management Domain. *Journal of computing in civil engineering,* 27, 36-50.

JALY-ZADA, A., TIZANI, W. & OTI, A. Building Information Modelling (BIM)—Versioning for Collaborative Design. Computing in Civil and Building Engineering (2014), 2014. ASCE, 512-519.

KARAN, E. P. & IRIZARRY, J. 2015. Extending BIM interoperability to preconstruction operations using geospatial analyses and semantic web services. *Automation in Construction,* 53, 1-12.

KLÉOS , J., RITA , A. & JOSÉ , C. IFC and the Representation of Reinforced Concrete Structures (Part 2)
An Approach to Load Combinations on Beams ANAIS DO 54° CONGRESSO BRASILEIRO DO CONCRETO - CBC2012 — 54CBC, 2012 Brasil. IBRACON.

KOCH, C. & FIRMENICH, B. 2011. An approach to distributed building modeling on the basis of versions and changes. *Advanced Engineering Informatics,* 25, 297-310.

MACDONALD, J. A. G., MARKO 2013. Multidisciplinary AEC Education Utilising BIM / PLIM Tools and Processes. *In:* KTH, S. O. A. A. T. B. E. A. (ed.) *PLM International Conference (PLM IC), PLM 13.* Nantes, France, 6th-10th July 2013.

NOUR, M. & BEUCKE, K. Object versioning as a basis for design change management within a BIM context. Proceedings of the 13th international conference on computing in civil and building engineering (ICCCBE-XIII), Nottingham, UK, 2010.

OTI, A. H. & TIZANI, W. 2014. BIM extension for the sustainability appraisal of conceptual steel design. *Advanced Engineering Informatics*.