Processes and Challenges to Develop Precast Concrete Test Cases for the National BIM Standard (NBIMS)

Samaneh Zolfagharian, samanehz@gatech.edu
Georgia Institute of Technology, USA

Prof. Charles M. Eastman, charles.eastman@coa.gatech.edu
Georgia Institute of Technology, USA

Abstract
The National BIM Standard (NBIMS) lays out the generic guidelines for developing specialized model views, defined in terms of Industry Foundation Classes (IFC). These model views are typically applicable to specific design and construction processes, and specific construction technologies. Various efforts have begun to develop specialized model views. One of these efforts is to create potential test models to check if conditions for passing a rule can be unambiguously identified. The aim of developing test models is to test each concept of one exchange model (EM) thoroughly and also combinations of different concepts within one EM. We tried both large comprehensive models and small ones testing only one or a few concepts. As well as good coverage, it is important to ensure that rules are not interfering with each other, so all test models should be checked against this set of conditions. If any error occurs during validating a test model, then it should be checked whether there is a bug in the test model or in rule logics. This paper reviews challenges in how to create test models, how to lay out a validation report, how to output a validation report so they strongly facilitate error correction, how to identify if a test model passes validation test. At the end of the validation process, a library will be provided for software companies to navigate and download needed files for undertaking detailed model tests. A user manual also should be considered that informs users how to walk through the testing processes.

Keywords: National BIM Standard (NBIMS), Model View Definition, IfcDoc Tool, Exchange Models

1 Introduction

The industry foundation class (IFC) schema is a standard for data interoperability (BuildingSMART, 2015b). The IFC model provides information about objects, processes, and relationships for sharing among architecture, engineering, and construction (AEC) during a project lifecycle. However, when representing this information, IFC models have some redundancies, so designers need a guideline for extracting information for various purposes and projects (Venugopal et al., 2012). To solve the redundancy issue and to enhance the interoperability process, the National BIM Standards (NBIMS) proposed model view definitions (MVDs) that enable users to define a subset of IFC schema for a specific purpose or project (BuildingSMART, 2015b). This paper reviews the process of developing a standard set of model views for precast concrete and the challenges of creating test models for validations. This project supports automated validation, which allows users evaluate the accuracy of BIM instance data with regard to an exchange model.

2 PCI-MVD Development Process

The NBIMs presented the procedural steps of developing model views that facilitate information exchanges. A holistic approach such as an MVD can provide a set of required functionalities for data exchanges (interoperability) so that the collaboration process improves among diverse systems and organizations. To develop an MVD, developers should understand the design or construction process being augmented and the required information for data exchange (Venugopal et al., 2012). This information can be classified by Business Process Modeling Notation (BPMN), which can describe the required process map, use-case, actors, and model exchanges (Venugopal et al., 2012). A use-case identifies information exchanges among multiple actors in a specific phase of a project (Venugopal et al., 2012). For example, Figure 1 shows the process map of a precast project in which architects are responsible for schematic designs in the preliminary phase. Once architects provide schematic designs, they will send an architectural concept model to the engineering group so that they can develop the engineering concept. Then the engineering group will send documents pertaining to the engineering concept to the architecture group for review.

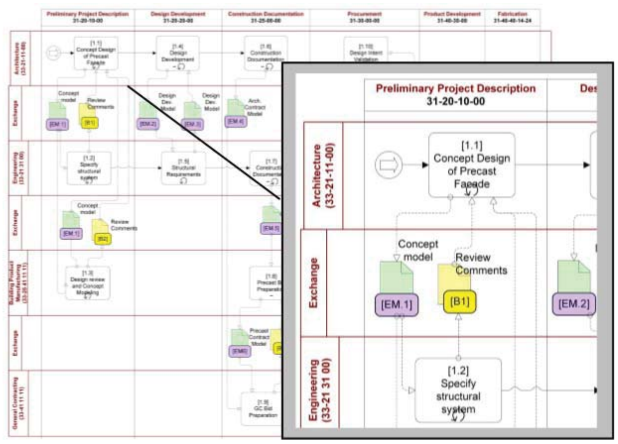


Figure 1 An example of a BPMN for exchanging information

After creating the BPMN diagram for information exchanges, we implemented the following phases for developing the MVD for the PCI project (Venugopal et al., 2012):

Phase 1: We captured detailed information for one or more use case exchanges and categorized this information based on an information delivery manual (IDM) for precast concrete. The IDM, a collaborative effort between the research technical team and the BIM Advisory Committee of the PCI, described a process map and exchange requirements. The IDM will be reusable for future user requirements. Moreover, we defined 51 exchanges.

Phase 2: We combined the exchange requirements identified in the IDM into a set of information modules called MVD concepts. These concepts are a modularized piece of the IFC used in specifying exchange requirements and implementation agreements subject to use for rule checking and debugging and reused by other exchange models (EMs). The PCI MVD includes 12 distinct EMs that address varying information needs for various data exchanges among five actors: an architect, a precast engineer, a precast detailer, a general contractor, and a plant manager. These EMs cover all project stages from the concept design to erection (see Figure 2). Figure 3 illustrates the relational structure of the PCI-063 concept, which shows IFC entities in the white boxes outside the blue box and the primary binding structure of a concept in the blue box.

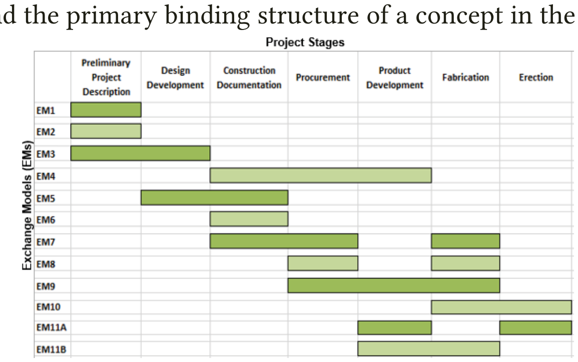


Figure 2 PCI exchange

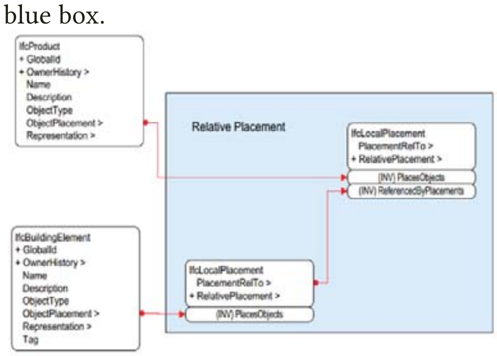


Figure 3 An example of a PCI concept documentation on the MVD page

Phase 3: We provided a set of guidelines demonstrating how to create object instances (e.g., beam/column, hollow core slab) in each EM. The guidelines list related concepts, attributes, and IFC entity mapping for each instance of implementation in BIM applications. Figure 4 illustrates related concepts, entities, and a list of mandatory attributes that are employed to create an instance of a complex slab. In addition, it shows the directions of definition for import and export, so designers can follow the concepts within tables for importing a part-21 file into a native model or exporting a part-21 file from a native model. For the start, EMs and the test models should be consistent.

Phase 4: We implemented MVDs and passed them to software companies that implemented test cases for related MVD concepts and validated the data import/export.

The research team at Georgia Tech did extensive work in developing a precast NBIMS following the described methodology. More details of these processes are available in (Eastman et al., 2011, Venugopal et al., 2012). This paper introduces the process of developing test models for PCI NBIMS and analyzes issues related to the semantics of PCI-MVD implementation.

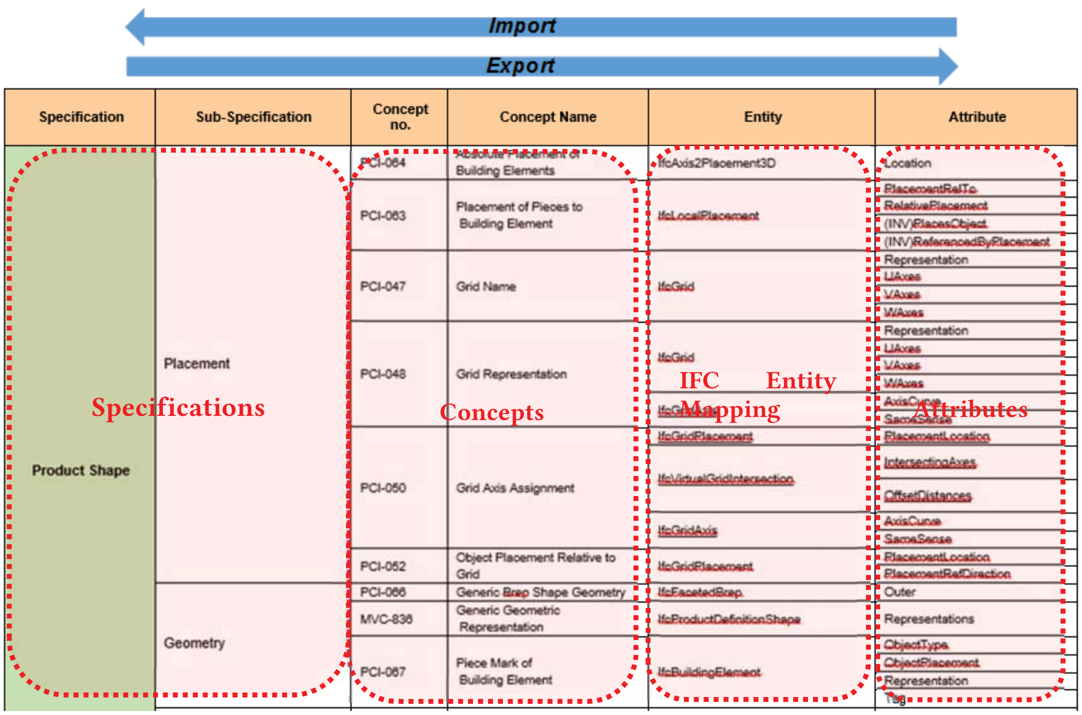


Figure 4 An example of the guidelines for building a test model

3 PCI Exchange Models

The PCI NBIMS includes 12 EMs covering all project stages from concept design to erection and addressing the requirements at various stages. The following section describes each EM and its geometry (DigitalBuildingLab, 2015). Because of the space limitation, we only include the definition of a few EMs.

EMPC1 - Building concept (BC): It consists of the concept design layout of precast pieces optionally composed of assemblies. Its geometry is nominal, without camber or twisting. It optionally includes major architectural finishes and site information. The geometry of this EM is extrusion.

EMPC2 - Precast concept (PC): It includes architects' and structural engineers' reviews of the building concept model. It specifies major architectural/structural precast components. The geometry of this EM is extrusion.

EMPC8 - Structural review and coordination (SRC): It includes geometry and assembly relations of buildings. It also includes detailed descriptions of precast piece detailing, all connections, finishes, joints, embeds, reinforcements, tensioning cable layouts and block-outs, pre-tensioned pieces, and hooks for lifting and transporting. The geometry of this EM is Brep.

EMPC9 - Engineering analysis results (EAR): It includes all structural precast elements. Important common categories of information include layouts, shapes, and material types. This exchange conveys the results of structural design and reinforcement review of the engineer of record and also the detailed fabrication model of precast pieces and assemblies provided by the precast fabricator. The geometry of this EM is extrusion.

EMPC10 - Final precast detailing and coordination (FPCD): It covers fully detailed information about products and their assembled composition in the project layouts, shapes, geometry, and finishes of all precast products prepared by a precast fabricator for coordination with precast and other systems, mostly by a contractor. The geometry of this EM is Brep.

EMPC11A - Production and erection data (PED): This exchange contains important common categories of information including layouts, shapes, material types, and product finishes both at the piece and assembly levels. The geometry of this EM is Brep

EMPC11B - Architectural review and coordination (ARC): This exchange contains the design constraints of buildings and spaces. Product information that raises issues about the design intent are reported, including layouts, shapes, material types, geometry, and the material of finishes both at the piece and assembly levels. The geometry of this EM is Brep.

4 IfcDoc Tool for Validation

Before providing guidelines for creating object instances, we created several comprehensive models that tested a large number of concepts and also simple models that tested only one or a few concepts. The test models consisted of object instances such as beams, walls, and slab elements. The aim of developing test models was to test every concept of one exchange model (EM) thoroughly and combinations of concepts within one EM. Besides covering a number of concepts in test models, we ensured that the rules did not conflict, so we validated all of the test models against sets of conditions using the IfcDoc tool developed by buildingSMART International (BuildingSMART, 2015a). The IfcDoc tool automatically generates IFC documentation for baseline IFC documentation and MVDs (Chipman, 2012). For validation purposes, the IfcDoc automatically generates a validation report in both IfcDoc and HTML formats. If any error occurs during validation, it can result from a bug in either the test model or the rule logic. Figure 5 shows an example of a validation report in IfcDoc. Our strategic goal was to eliminate all rule logic errors. For further explanation about the validation process using IfcDoc, refer to (Yong-Cheol Lee et al., 2015).

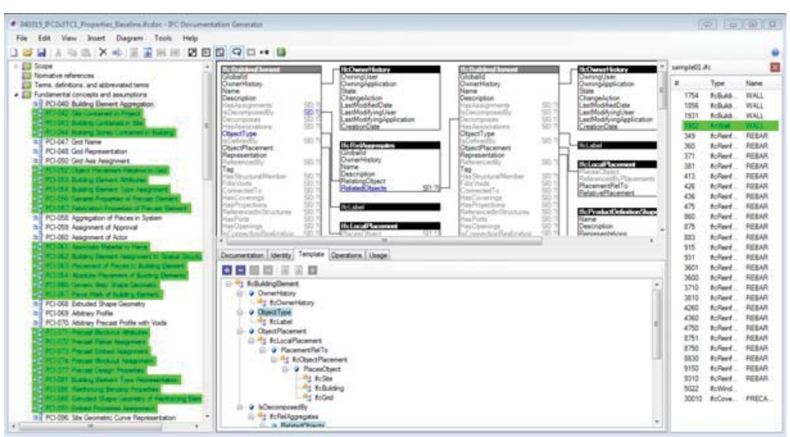


Figure 5 An example of validation report in IfcDoc tool

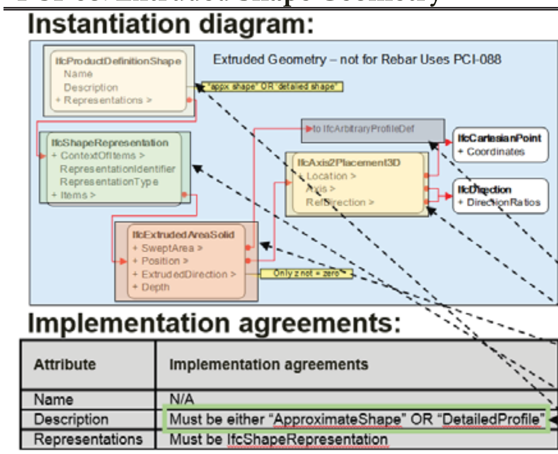
5 Strategies and Challenges for Building PCI Test Models

This section discusses the strategies and the challenges of creating and validating test models for PCI-MVD. The research team provided a library that software companies can use to navigate and undertake detailed test models. The precast concrete concept structure was developed in the 2009 timeframe, before new technology was developed to automatically identify concepts. The concepts were defined and composed by hand. The research team manually created text-based instance models in the IFC format. The created test models satisfy all of the concepts and predefined rules in the MVD. However, no application is able to generate test models in this manner, so the research team had a big challenge to create test models manually. They had to have enough knowledge and understanding about IFC2x3 documentation, concept definitions, and concept implementations.

To create the test models, we first selected a project stage and relevant EMs shown in Figure 2. Each EM includes a list of concepts that provides detailed specifications. Each PCI concept contains a definition of the concept, an instantiation diagram of IFC bindings, and implementation guidance for software developers (V. Aram et al., 2010). Table 1 illustrates an example of the "PCI-068: extruded shape geometry" concept, in which the instantiation diagram shows the relational structure of the concept. In addition, the "implementation agreement" section describes some predefined values and rules, including logic operators such as "AND" and "OR."

Table 1 An example of mapping MVD concept (PCI-068)

PCI-68: Extruded Shape Geometry	Part #21




```
#1084=IFCCARTESIANPOINT((15.56385,4.2672,17.9832));
#1085=IFCAXIS2PLACEMENT3D(#1084,#5,#3);
#1086=IFCLOCALPLACEMENT(#16,#1085);
#1088=IFCCARTESIANPOINT((0.3556,-0.6096));
#1089=IFCCARTESIANPOINT((0.4953,-0.6096));
#1090=IFCCARTESIANPOINT((0.508,-0.3048));
#1091=IFCCARTESIANPOINT((0.3556,-0.3048));
#1092=IFCCARTESIANPOINT((0.3556,0.));
#1093=IFCCARTESIANPOINT((0.,-0.6096));
#1094=IFCPOLYLINE((#1088,#1089,#1090,#1091,#1092,#46
9,#1093,#1088));
#1095=IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,'LBE
AM<LB-1><BEAM>',#1094);
#1096=IFCAXIS2PLACEMENT3D(#2,#3,#466);
#1097=IFCEXTRUDEDAREASOLID(#1095,#1096,#5,14.592
3);
#1098=IFCSHAPEREPRESENTATION(#7,'Body','SweptSolid
',(#1097));
#1099=IFCPRODUCTDEFINITIONSHAPE('','ApproximateShape',(#10
98));
#1083=IFCBEAM('1wTkxpU315SA_Ba91qS1D5',#8641,'LBEAM-
2<LB-1>','LB-1',$,#1086,#1099,'TS 9201');
```

The second step in the process of creating the test models was to simplify them. We classified the concepts within each EM into four categories: placement, geometry, relations, and property sets. Then we followed the IFC binding table, values, and rules in each concept document to create an object instance. For example, to create the geometry of a beam functionally needed in EM9, we first identified related concepts to define the type of the geometry extruded in EM9 and then followed the instantiation diagram of the concept. Table 1 shows how we mapped instantiation diagrams and rules into a part 21 file (i.e., an IFC format). In this step, we also included various methods of generating models for some specific building elements such as slabs and sandwich walls. For example, designers can design a slab instance in various representations depending on the required level of details for the element, so we created two types of slab instances for different purposes: a complex slab instance and a simple slab instance. A complex slab is created by the aggregation of beam elements (IfcBeam instances) and topping (IfcBuildingElementpart instances); however, a simple slab element is created directly as an IfcSlab instance. Figure 6 shows an IFC binding diagram for creating a complex slab instance. In the test model for a complex slab, we created an element assembly instance and assigned it to a beam instance in the complex slab. Then, to create an aggregation instance of beam instances, we used IfcRelAggregation. We also used IfcRelAggregation to create an aggregated instance of beam assemblies with topping, reinforcements, and embeds. To sum up, a complex slab is the aggregation of beam elements, that is, a hollow core plank or a double tee, and topping.

The next step of the process was validating the created test model to ensure that we accurately implemented all of the rules, values, and relationships. However, we found several errors that had occurred during the validation process, so we debugged them before releasing the PCI-MVD for use by software developers. Debugging included some changes in the definition of the concepts, rules, or

IFC bindings. In addition, we discovered several errors resulting from the limited functionalities of IfcDoc for handling all of the rules, values, and relationships.

We created both large comprehensive and small test models. Creating small test models were easier for us because we were testing only one or a few concepts. However, creating a comprehensive test models were so challenging because we found some redundancies in definitions of some concepts, so we merged or removed them from the MVD. In addition, definition of some concepts conflict with each other, so we modified those definitions to fix the issue. Since we manually created test models, creating a comprehensive model was so time consuming. Moreover, tracking all instances and their relationship in part 21 files which usually contain more than 3000 instances was an arduous task.

A comprehensive test model usually consists of multiple assemblies such as element assemblies or reinforcement assemblies. Since assemblies were created by IfcElementAssembly, we had a big challenge to differentiate element assemblies from reinforcement assemblies during validation process. For example, one issue that we were struggling with was related to the "PCI-103: aggregation of reinforcing assemblies" and the "PCI-104: aggregation of reinforcing element to reinforcing assembly" concepts. Concept PCI-103 pertains to assigning an aggregation of reinforcement elements to a higher level of composition such as a rebar cage, so the instantiation diagram of this concept contains relationships between two IfcElementAssembly and IfcRelAggregate. However, if a test model included this type of relationship (IfcElementAssembly and IfcRelAggregate) among other building elements except reinforcement elements, then IfcDoc mistakenly validated those instances against the PCI-103 and PCI-104 concepts. IfcDoc was not able to recognize the type of the relationships because these two concepts did not include any predefined generic type for the objectType attribute in IfcElementAssembly. Thus, these concepts were applicable to not only reinforcement assemblies but also all other building element assemblies. To address this issue, we assigned a predefined type (=Rebar) to the objectType attribute in IfcElementAssembly in the PCI-103 and PCI-104 concepts so that we could differentiate reinforcement assemblies from element assemblies in the validation process. For instance, Figure 6 illustrates multiple element assemblies including element assemblies and reinforcement assemblies, but IfcDoc should only validate the reinforcement assembly against the PCI-103 and PCI-104 concepts. In this example, IfcDoc can validate the beam assembly against the "PCI-040: building element aggregation" concept and not the PCI-103 and PCI-104 concepts.

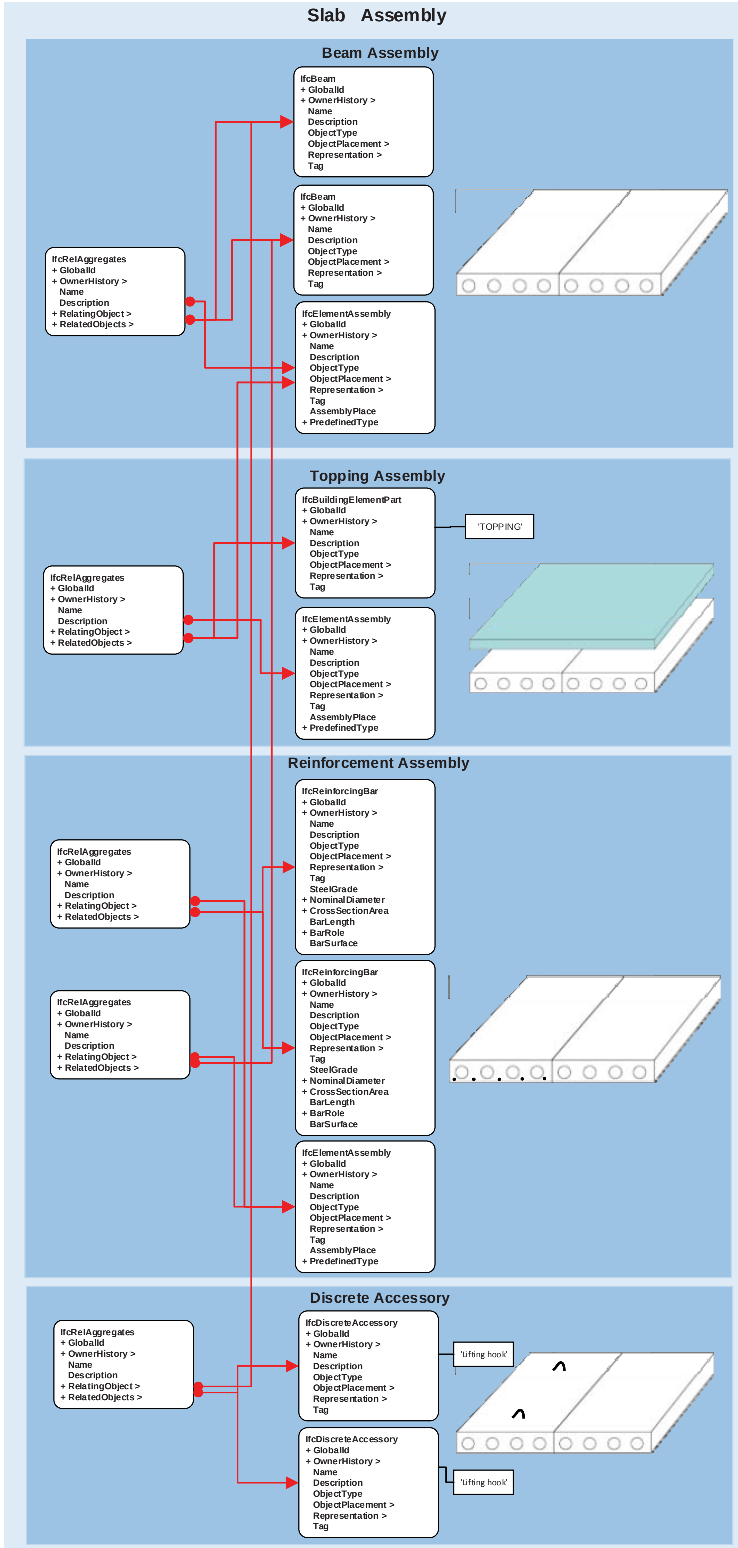


Figure 6 IFC entities and their relations for creating a complex slab

6 Lessons Learned and Path Forward

Creating test models assists the research team to identify redundancies and conflicts among concepts in PCI-MVD. The research team performed a huge task to avoid redundancy and rework in terms of the development and testing of MVD Concepts. We merged or removed those concepts that were redundant in the definition of concepts or Ifc binding diagrams. We also added some rules to those concepts that were interfering with each other so that we could prevent any possible confusion during the validation process. For example, we assigned a generic type to objectType attribute in the PCI-103 and PCI-104 concepts so that we could differentiate element assemblies from reinforcement assemblies.

Creating test models also assists us to improve capabilities and functionalities of IfcDoc tool. Each time that we created a test model and validated it, we could identify a new bug or a limitation in IfcDoc. Then, we reported them to IfcDoc developers so that they addressed these limitations and released a new version of IfcDoc tool. For instance, IfcDoc developers added the color-coding functionality to IfcDoc that helps users to understand and analyze validation reports more efficient. Figure 5 shows an example of color-coded validation reports in which green color means pass and red color means the test model did not satisfy the concept. For more detailed review of validation reports, refer to (Yong-Cheol Lee et al., 2015).

7 Conclusion

IFC is a rich model that addresses the needs of different applications and provides a variety of ways to define the same part of a building. Hence, designers require additional layers of specificity such as model views for IFC implementations. This paper describes the process of developing an MVD for precast concrete as well as challenges in creating test models. The aim of creating test models was to provide a better understanding of PCI-MVD for designers. In addition, it helped us find if there is any redundancy in definition of concepts. We could also identify those concepts that were interfering with each other.

Beside test models, we created some guidelines for generating a native model that supports a specific EM for translation, with the set of concepts provided in PCI Model Building documentation. Designers can follow the concepts within tables for importing a part-21 file into a native model or exporting a part-21 file from a native model. Currently, software developers implement EMs in their IFC translators, debug their translator based on the MVD, and learn trouble spots and debugging guidelines.

Acknowledgements

The work presented here was funded by the Prestress/Precast Concrete Institute. All information presented is that of the authors alone.

References

BUILDINGSMART. 2015a. *ifcDoc Tool Summary* [Online]. [Accessed April, 4 2015].

BUILDINGSMART. 2015b. *Industry Foundation Class (IFC) data model* [Online]. http://www.buildingsmart.org/standards/technical-vision/open-standards-101/. [Accessed April, 4 2015].

CHIPMAN, T. 2012. IFCDOC Usage Guide https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&uact=8&ved=0CCYQjBAwAw&url=http%3A%2F%2Fwww.buildingsmart-tech.org%2Fspecifications%2Fspecification-tools%2Fifcdoc-tool%2Fifcdoc-help-page-section%2FIfcDoc.pdf%2Fat_download%2Ffile&ei=_IoiVe3BK4XdsAX8vIGoAw&usg=AFQjCNHaaiWeBrejhjnEL4leFe-KRDTMZg&sig2=ZBXe4vd_zpnQDNmLklGVaQ: ifcDoc Tool Summary — Welcome to buildingSMART-Tech.org.

DIGITALBUILDINGLAB. 2015. *Precast Concrete BIM Standards* [Online]. [Accessed April, 21 2015].

EASTMAN, C., PANUSHEV, I., SACKS, R., VENUGOPAL, M., ARAM, V., SEE, R. & YAGMUR, E. 2011. A guide for development and preparation of a national bim exchange standard. *buildingSMART Report.< http://www. buildingsmartalliance. org/client/assets/files/bsa/IDM-MVD_Development_Guide_v4. pdf>(Jan. 09, 2012).*

V. ARAM, C. EASTMAN, R. SACKS, I. PANUSHEV & VENUGOPAL, M. 2010. Introducing a new methodology to develop the information delivery manual for AEC projects. *CIB W78*, 16-18.

VENUGOPAL, M., EASTMAN, C. M., SACKS, R. & TEIZER, J. 2012. Semantics of model views for information exchanges using the industry foundation class schema. *Advanced Engineering Informatics,* 26, 411-428.

YONG-CHEOL LEE, TIM CHIPMAN & EASTMAN, C. M. The Validation Logic and Structures of a Building Information Model Pertaining to the Model View Definition. Proc. of the 32nd CIB W78 Conference 2015, 27th-29th 2015, 2015 Eindhoven, The Netherlands. (Under review).