



CHARLES PANKOW
FOUNDATION
Building Innovation through Research



A GUIDE FOR DEVELOPMENT AND PREPARATION OF A NATIONAL BIM EXCHANGE STANDARD

By

Chuck Eastman

Ivan Panushev

Rafael Sacks

Manu Venugopal

Shiva Aram

Richard See

Version 2.0

6-25-2011





Table of Contents

PREFACE	4
0. OVERVIEW	7
1. PHASE 1: STANDARDS REQUIREMENTS	12
1.1. BIM STANDARD SCOPING.....	12
1.2. WORKGROUP FORMATION.....	13
1.3. SCOPE OF NATIONAL BIM STANDARD.....	15
1.4. EXTERNAL REPORTING TO BUILDINGSMART.....	16
1.5. PROCESS MAP.....	16
1.6. DEFINING THE PROCESS MAP.....	21
1.7. DEFINING EXCHANGE REQUIREMENTS AND BUSINESS RULES.....	22
<i>Exchange Descriptions</i>	22
<i>Exchange Requirements:</i>	24
<i>Business Rules:</i>	27
1.8. THE INFORMATION DELIVERY MANUAL.....	30
1.9. GENERIC BIM USER GUIDE.....	30
1.10. IDM SUBMISSION OF BUILDINGSMART NORTH AMERICA.....	31
2. PHASE 2: DESIGN	32
2.1. MODULARIZATION OF MODEL VIEWS.....	32
2.2. EXCHANGE REQUIREMENT MODEL.....	33
2.3. A GUIDE TO IFC SOLUTIONS FACTORY.....	36
2.4. DEFINITION OF CONCEPTS FOR UPLOADING.....	44
<i>Defining the IFC Bindings in Static Concepts</i>	44
<i>Defining the Adapter Concepts</i>	47
<i>Review of Draft MVDs</i>	48
<i>Relating the Developed Concepts to IDM Exchange Requirements</i>	48
2.5. UPLOADING THE MVD AND CONCEPTS.....	49
3. PHASE 3: CONSTRUCT	52
3.1. FACILITATE SOFTWARE PRODUCT IMPLEMENTATION.....	52
3.2. UNIT TESTING AND GENERAL TESTING.....	53
<i>Development of Test Files</i>	54
<i>Concept Level Testing</i>	55



Aggregated testing:..... 62

Implementer’s Conference Calls: 63

3.3. SW CERTIFICATION TESTING AND REPORTING 63

3.4. SUBMISSION OF A DRAFT MVD FOR REVIEW..... 64

PHASE 4: DEPLOY 65

3.5. PRODUCT SPECIFIC BIM GUIDES 65

3.6. BIM EXCHANGE AND DATA VALIDATION 65

4. POST SCRIPT 66

5. GLOSSARY 67

6. BIBLIOGRAPHY 68

APPENDIX A: SAMPLE IFC BINDING DOCUMENT 69

APPENDIX B: PRECAST JOINT MODEL EXCHANGE VALIDATION 73

APPENDIX C: SAMPLE PART-21TEST FILE 77

APPENDIX D: LIST OF SUPPORTING DOCUMENTS FOR MVD DEVELOPMENT 85

APPENDIX E..... 86

Preface

This "How To" manual is a guide for those planning to develop Information Delivery Manuals and/or Model View Definitions generally following the processes defined in the US National BIM Standard. An **Information Delivery Manual (IDM)** defines one or more Exchanges of BIM information in the context of reference industry processes. IDMs are defined by end users and practicing professionals and serve as the Requirements Definition for such BIM exchanges. A **Model View Definition (MVD)** is defined by the buildingSMART organization as "a subset of the IFC schema that is needed to satisfy one or many **Exchange Requirements** of the AEC industry." (see http://www.iai-tech.org/products/ifc_specification/ifc-view-definition). In fact, an MVD can be configured to enable the information exchange using any product model schema, not only for the IFC product model. A more generic definition of an MVD is therefore "a subset of a building product model schema that provides a complete representation of the information concepts needed for a particular information exchange in an AEC workflow."

The process of defining standardized BIM information exchanges has evolved and been refined over the last 4 years. Until 2008, there were two separate development teams – developing IDM and MVD separately. At that time IDM developers mapped functional specifications directly to the IFC schema and the MVD developers began their process by developing a Generic MVD (similar to the Exchange Requirements Model (ERM) defined in the next section). In late 2007, the IDM and MVD teams agreed to integrate their processes in the manner described in this document. Primarily the agreement meant that IDM is focused on end-user requirements definition and MVD is focused on the translation of those requirements into exchange representations that can be implemented in software products for use in AEC industry projects. IDMs are defined in the form of Process Maps and exchange requirements. MVDs are documented in the form of an Exchange Requirements Model and Model View

Definitions with associated implementation guidance documents – one per MVD Concept.

This document reports current best practices and provides a step-by-step guide. The development of one or more IDMs and MVDs is a major undertaking, usually requiring inputs from dozens of people and taking multiple calendar years to complete. The purpose of the MVD is to automate the exchanges within the current set of workflows in some AEC industry processes. It is important to recognize that the greatest benefits of Building Information Modeling and interoperability arise out of developing improved workflows. Thus MVDs automating current workflows change the context of workflows and uncovers new workflow options, leading to MVD revisions. MVDs are live, and will evolve over time.

The examples presented here come from the experience gained by the authors through preparation of an information delivery manual (IDM) and a set of MVDs for the precast concrete domain. It also incorporates and builds upon the experience and guidelines for defining Information Delivery Manuals and Model View Definitions developed by the BLIS Consortium, buildingSMART, and Digital Alchemy.

The MVDs addressed here are those that might be encountered throughout the building lifecycle, although most of our experience to date addresses mostly design, engineering and construction aspects of the full lifecycle. Our experience has emphasized the made-to-order aspects of construction, such as structural systems and cladding. Other examples for exchanges ranging from Spatial Program Validation, to Energy Analysis, to Quantity Takeoff, and Information Handover to Facilities Management can be reviewed on the IFC Solutions Factory web site at: <http://www.blis-project.org/IAI-MVD/>.

The document is organized roughly in an outline structure, with major topic headings of the high-level parts of the process, then within those sections are elaborated the details for realizing them.

For the sake of continuous improvement, we invite comments on this report and suggestions on how to improve it, as others gain experience in compiling IDMs and MVDs.

0. Overview

The purpose for developing a national BIM standard is to provide a layer of specificity over the top of an IFC or other exchange schema. In the case of IFC, and possibly other schemas, the schema is highly redundant, offering multiple ways in which a concept can be exchanged between applications. The purpose of a BIM standard is to select and specify the appropriate information entities from a schema for particular use cases (or data exchange scenarios). The selected entities that comprise a model view definition are a subset of all those in the schema.

A BIM standard is organized around one or more Use Cases. A *Use Case* addresses the exchange needs of two actors (say architect and fabricator) at a given stage in the building life-cycle. A Use Case identifies at least one information exchange. Quite often, it defines a dialogue, involving multiple exchanges. An *information exchange* defines the data that must be specified to support one exchange from the end-user's perspective. It identifies the entities, relations, properties and other data needed by an information user and supplied by an information provider.

The process presented here generally follows the procedures set forth in The **National BIM Standard™ Version 1 Part 1**. (The Standard is downloadable from <http://www.buildingsmartalliance.org/projects/products.php>.) Section 5 of the National BIM Standard outlines the procedural steps to be followed. [This document also draws upon the ISO 29481-1-2010 documentation on IDM and the Integrated IDM-MVD Process Formats.](#)

This document presents a draft process, as it was published before the process had been extensively tested and before the supporting organizational structures were put in place. Thus it is only an initial guide. This document, on the other hand, offers a practical set of guidelines that have been tested and followed, with known outcomes in various projects in buildingSMART and by other organizations mandating or encouraging IFC data exchange (e.g. the US General Services Administration, the Charles Pankow Foundation, and the Precast Concrete Institute).

The NBIMS process is shown in Figure 1. It outlines thirteen steps in four major phases. These steps will be referenced and in some cases elaborated (or modified) throughout this document. Here we provide a three page overview of its structure and steps:

(1) Forming a workgroup and identifying the scope and context for one or more use case exchanges. The workgroup is composed of experts in information technology applications and information exchange for AEC and also experts in the construction domain being considered. The context is defined through compilation of one or more process maps that identify where the exchanges take place in the project lifecycle and the actors and applications that are the senders and recipients of the exchange(s). For each exchange, the functional requirements of the information to be exchanged are defined, called Exchange Requirements. Exchange Requirements are typically defined in a table or spreadsheet. These two things (process maps and exchange requirements) are combined to form an Information Delivery Manual (IDM). The IDM serves as the overall requirements specification for one or more exchange(s). The IDM may be submitted to bSa separately for review and

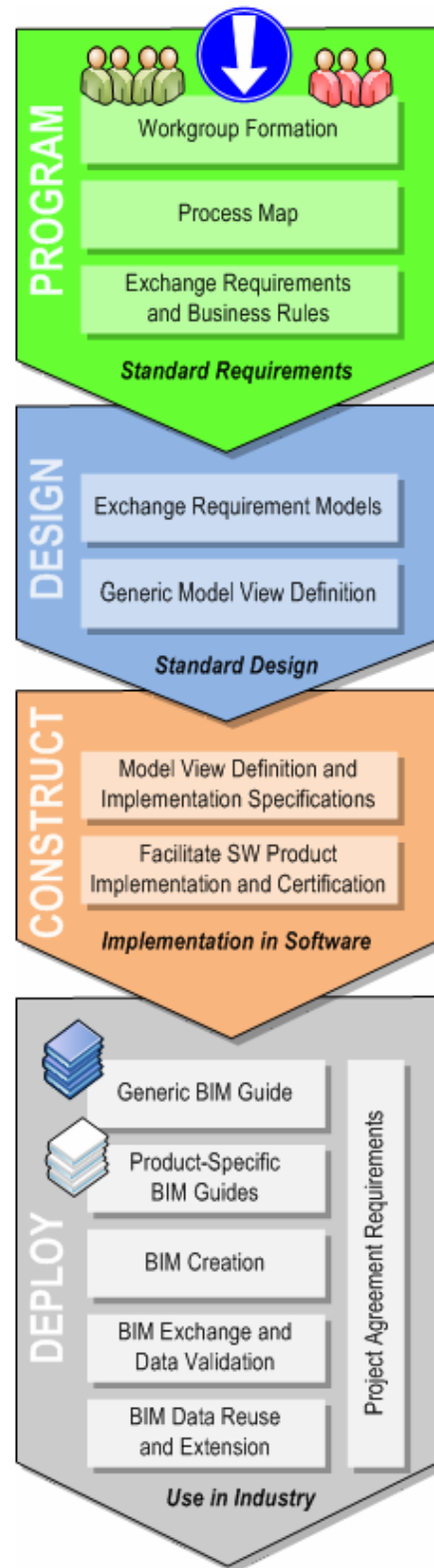


Figure One: NBIMS Process

referencing.

(2) The data elements defined in the Exchange Requirements Model diagrams are next structured into a set of information modules. These information units of the exchange are called Concepts. A Model View is defined as a collection of such Concepts, which will later be mapped to the implementation schema (IFC most commonly) – and is called a Model View Definition (MVD). The input of software developers, representing the major vendors in the domain, is highly desirable in this point in the development process. MVD Concepts are modular and intended to be reused across multiple MVDs. In this way, there can be a single definition, implementation (per product), and test configuration for the concept – which is used in many exchanges. Concepts are shared through an open website, IFC Solutions Factory, at <http://www.blis-project.org/IAI-MVD/>. Developers of new MVDs are encouraged to use any Concept which already exists, rather than creating new Concepts. The advantage is the documentation and testing has already been developed and it is simply used in the new MVD. In exchange for this open reuse, the new MVD developer must agree that new Concepts they document will be added to the open collection of Concepts – available for reuse in other MVDs.

A well-structured set of templates has been developed for documenting Concepts and their aggregation into higher-level Concepts, then into an MVD which supports one or more exchanges. When the templates are filled out, the resulting online documentation serves as the specification for the MVD, which is the second major document in developing a BIM standard. Tools have also been developed to validate an MVD relative to the requirements in the IDM for which it has been developed.

(3) The third phase addresses the implementation of MVDs by software companies and certification of those applications for correct implementation, as defined in the MVD. Certification is assessed through a detailed and exhaustive testing and reporting process. This testing is done MVD Concept by Concept. The Concept based approach to testing has benefits over previous test approaches in the following ways: (a) test

software and test case development is made efficient, as they can be reused for certification of all MVDs that include the same concept; (b) software implementation is made efficient because a single implementation can be used in multiple exchanges and the vendor is assured that, if it passes for one exchange, it should pass for the other exchanges.

Administration of software implementation and certification testing will vary by MVD project (generally funded directly by the MVD development team or sponsors), but these principles should be observed in all cases. Test sites, developed to support certification, are being readied. High confidence in reliable exchange of design and engineering data is the target outcome. The MVD and the IDM it was based on (if not submitted earlier) are submitted for review and referencing to bSa.

(4) The last stage of the NBIMS process is deployment of certified applications and use of those applications in AECO projects. This starts with the development of product specific BIM Guides which tell the user how to implement the industry processes and exchanges defined in the IDM using the particular software product.. This will allow the users of applications to prepare models suitable for the needed exchanges. This phase also includes the development of building models requirements that will enable the target IDM exchanges in industry projects. Case studies are expected, with assessment of the outcomes resulting from previous stages. The overall process is meant to incorporate best software engineering practices.

Another important aspect of success in the deployment phase is the availability of third party BIM Data Validation services. These will enable end users to upload building models and validate conformance to requirements in a selected IDM/MVD. This capability will support the application of contractual requirements specifying standard BIM exchanges; this capability will allow both sending and receiving parties in an exchange to validate if all data exchange requirements have been satisfied.

Throughout this process, support is provided by meetings (physical or online) that communicate and review the issues arising at each step.

The following sections provide detail to each of the phases and steps.

1. PHASE 1: Standards Requirements

Phase 1 Overview: This phase organizes the team that will participate in defining the Model Views, defines the scope of the endeavor, and determines the functional requirements for the exchanges. It is the fundamental step that defines the functionality of exchanges to be supported by MVDs.

1.1. BIM Standard Scoping

The scope of a BIM standard can vary tremendously. In some cases, the Use Cases and exchanges are meant to cover the exchanges within an industry domain over one or a range of design and construction life-cycles. This was the case for the domains of precast concrete. These cases typically involve many different Use Cases and information exchanges, addressing a wide and varied range of information. We call this an *Integrated Use Case* approach. In other cases, the exchange addresses a single or small number of Use Cases with only one or two exchanges, for example for building model data input into an energy analysis in late concept design, or for handover from contractor to facility manager at the end of construction. These may only be one-way and one single exchange (possibly with iterations). We call this the *Basic Use Case* approach.

This document aims to serve as a guide for both types of Use Cases. A couple of considerations: the value of the Integrated Use Case approach is that the general definition of the information entities for the whole domain can be defined initially in a consistent fashion, and then incrementally implemented. This is important if a Basic Use Case is expected to grow into many Use Cases in the future. The Basic Use Case, on the other hand, leads to quick results and implementation. A mix of the two approaches is also possible and may be the best. Scoping to address the most critical workflows over the lifecycle, with simple aggregated exchanges, may allow both broad implementation and quick benefits (although important ones) to a wide set of users.

An important consideration is that the targeted Use Cases can be implemented within current IFC schema capabilities. Use Cases that require extensions to the IFC model schema are not uncommon, but should not be initiated without full disclosure of the time scale implications. The cycle time for update of the IFC schema to incorporate new entities, properties and relationship is three to four years. The most recent upgrade was published in November 2010. Use Cases, especially early ones, should therefore rely on the current production release of the IFC schema. Requirements for a schema extension will dramatically change the schedule and focus of work. If extensions are required as part of an NBIMS specification,, the extensions should be undertaken in parallel to other NBIMS activity, keeping their capabilities separate from Workgroup IDM and MVD activities until the extensions have been adopted and built into an IFC release.

1.2. Workgroup Formation

Different parts of the building industry have different information exchange needs. For example, the information need of architects is vastly different from those of building component fabricators of different kinds. People with similar needs are most commonly in the same profession. The need for interoperability should be discussed and agreed upon in order to initiate a workgroup. The workgroup arises out of that shared need. Team membership generally includes domain professionals from the user community with a vested interest in improving interoperability in the their domain.

The domain Workgroup is made up of three types of actors: a user group representing the domain, a technical advisory group to support the process and implementation, and a software group of expected implementers of the exchange software. The domain user group provides the construction industry expertise to identify the exchanges that need to be supported. The construction industry has many professional organizations that serve different constituents – AIA, AGC, AISC, ACI, etc. To date, these organizations have provided administrative leadership for many of the NBIMS initiatives. An ad hoc

group that has a common interest can also become a workgroup; especially if they are a part of a larger umbrella organization.

One or more technical advisors support the process. Typically these advisors have been through the IDM/MVD development process previously. They will have familiarity with the software tools used to develop the specifications, and should be effective in working with the other groups in the process. They should also have experience working with the technical organizations dealing with validation, testing and certification.

A partial listing of potential technical advisor contacts is given in Appendix E. This group typically is comprised of consultants or university faculty, etc. and takes the lead in organizing meetings, providing leadership regarding the process. These activities can be paid for by an industry group or member contributions for this activity. Some BIM standard groups are currently attempting to use user group company staff for this support. Beside employee time contributions, the hiring of technical advisors is the major expense of the IDM/MVD development process.

Undertaking an NBIMS IDM/MVD development project requires careful consideration. It involves significant commitments of time and expertise. Applications that are envisaged to exchange information must already be in practical use. The user community should already be at the point of using BIM, so that there is a recognized problem regarding data exchange. If the initiative is started too early, motivation will be lacking and the effort will be harder to realize.

Guidelines for workgroup formation:

1. The workgroup should consist of between 5 and 15 industry technical members. The members need to know the business well, its procedures and daily practice. Members should represent a cross section of the industry it represents.
2. Members are expected to make in-kind contributions for their efforts, paid for by their employer. If the work is not to drag over several years, the expected range of commitment is 5-10 hours per week. The members will be expected to

- discuss, organize process model information, identify information needs of the different process exchanges they identify, and provide discriminating detail about the information needed.
3. The workgroup needs to elect or appoint a chair. The chair is the leader of the working group, organizing meetings, serving as a spokesperson for the project, encouraging company executives in the domain of the importance of the workgroup's activities and promoting company support for the activity through assignment of their often most valuable staff. The in-kind costs of developing a BIM standard are easily in the tens of thousands of dollars per company.
 4. Representatives from both sides of the information exchange(s) must be included and active in the team.

The NBIMS workgroup, its chair, technical advisors, domain advisors and software company representatives should outline a general schedule of physical meetings, at least two per year, with in-between conference calls. A website for the NBIMS facilitate communication and report and file distribution is very desirable. A typical NBIMS project will typically require two to three years, if we include implementation by software developers and testing.

The forming of an NBIMS Workgroup is reported to the National BIM Standard Planning Committee. It identifies the general scope, its members, chair and technical advisors. A template for this information is available from the NBIMS website.

1.3. IDM/MVD Project Scope

A strategic issue to be decided in the initial meetings is the scope and schedule of the effort. The scope can vary, ranging from defining a single Use Case to a set of Use Cases, around some function or role, such as steel fabrication or energy analysis. If there is an expectation that the domain group will require a number of diverse Use Cases, then they are most economically addressed together, in an Integrated Use Case approach. This is efficient in the long run, but can extend the development time because of its scope. The advantage is that the Workgroup of users, technical advisors

and software implementers, can develop a consistent and efficient approach to deal with the issues within the domain. But even in this situation, the implementation should be broken down into phases, so that the Workgroup and outside members of the domain user community will see quick and visible progress.

1.4. External Reporting to buildingSMART

After the formation of the Workgroup, its existence should be reported to the buildingSMART Project Committee. This reporting should define its membership, and most importantly its scope. The buildingSMART Project Committee will report back any previous or current NBIMS Workgroup activities that bear relation to the one submitted and provide communication links with the appropriate parts of the NBIMS community.

1.5. Process Map

In order to define Use Cases, NBIMS has adopted the convention of most such efforts that identify the context of an exchange, by placing it in a process map. The process map identifies when in the building lifecycle the exchange takes place and the roles of the people sending and receiving the data. An example process map is shown in Figure 2.

NBIMS has adopted Business Process Modeling Notation (BPMN, www.bpmn.org), as the process representation tool. There are multiple BPMN diagramming tools. Visio, which has a plug-in for BPMN shapes (see the BPMN website, at <http://www.bpmn.org/documents.htm>), was used to prepare the process maps shown in Figure Two. Alternatives are listed on the BPMN website and Visio based stencil and template are also provided in the NBIMS Team Briefing Kit. The BPMN naming conventions followed are explained in Figure 2 and an example process map for structural precast is shown in Figure 3.

The horizontal and vertical rows in a BPMN diagram, called “swim-lanes”, are used to categorize activities with different functional objectives or capabilities. .The horizontal

swim lanes identify the actors and the context of the activities through the chronological progression through the lifecycle of a business process. The vertical swim lanes delineate the process into distinctly named stages. Referring to Figure 3, the conventions for defining the phases of a building construction business process map are categorized using the CSI Omniclass, Table 31, Phase, classification. The Omniclass definitions are available from: <http://www.omniclass.org/>. These are defined in the column headers across the top of the sheet and are meant to identify when in the project lifecycle the Use cases are targeting.

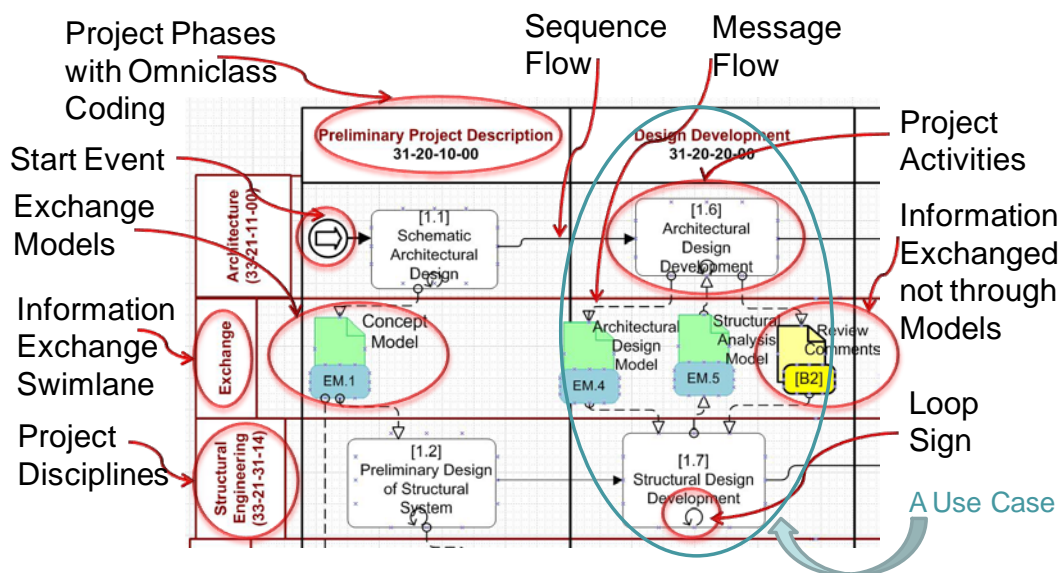


Figure 2: Illustration of naming conventions followed in developing process maps

The rows of the process map are the relevant actors or roles involved in the exchanges. These are also defined in OmniClass categories, using Table 33, Disciplines, classification. The Domain group may have to approximate the disciplines or roles, as the OmniClass categories do not yet cover all construction industry related roles.

In between the Discipline rows there are “Exchange” rows. These organize and group exchanges between Disciplines. At the Process Map level, white rectangles with rounded corners signify Activities, located within the appropriate Discipline’s row and project phase column. Each has an identifier, linking it to a more extensive description

of the task. Within an Activity box, there may be several symbols across the bottom; a directed arc designates the Activity may be iterated. A plus box indicates the Activity is a high level description made up of a set of design Activities described separately and hierarchically – BPMN provides hyperlinks between high-level and detail Activities. (The full graphic syntax of BPMN is available from <http://www.bpmn.org/>). The corner folded blocks in the Exchange lanes designate an information exchange. The green information exchanges are building model exchanges, while the yellow ones represent all non-building model exchanges (such as tables of data, documents, or verbal communication). The exchanges also have IDs for cross referencing. The building information model exchanges, represented by the green symbols, are the focus of our interest.

The dotted lines denote information flows from an Activity to a Use Case exchange (export) and hence to the receiving Activity (import). Branching within a flow implies two or more targets of the flow. By definition, a Use Case is an exchange between different Disciplines and thus cannot be between two tasks in the same Discipline swimlane. (If a Discipline uses multiple software, then these should be broken out as separate swimlanes.) Solid lines show flow-of-control relations between Activities within a single Discipline swimlane; they should not cross between Disciplines.

Figure 3: shows an example of the process maps developed following the methodology mentioned in the previous paragraphs.

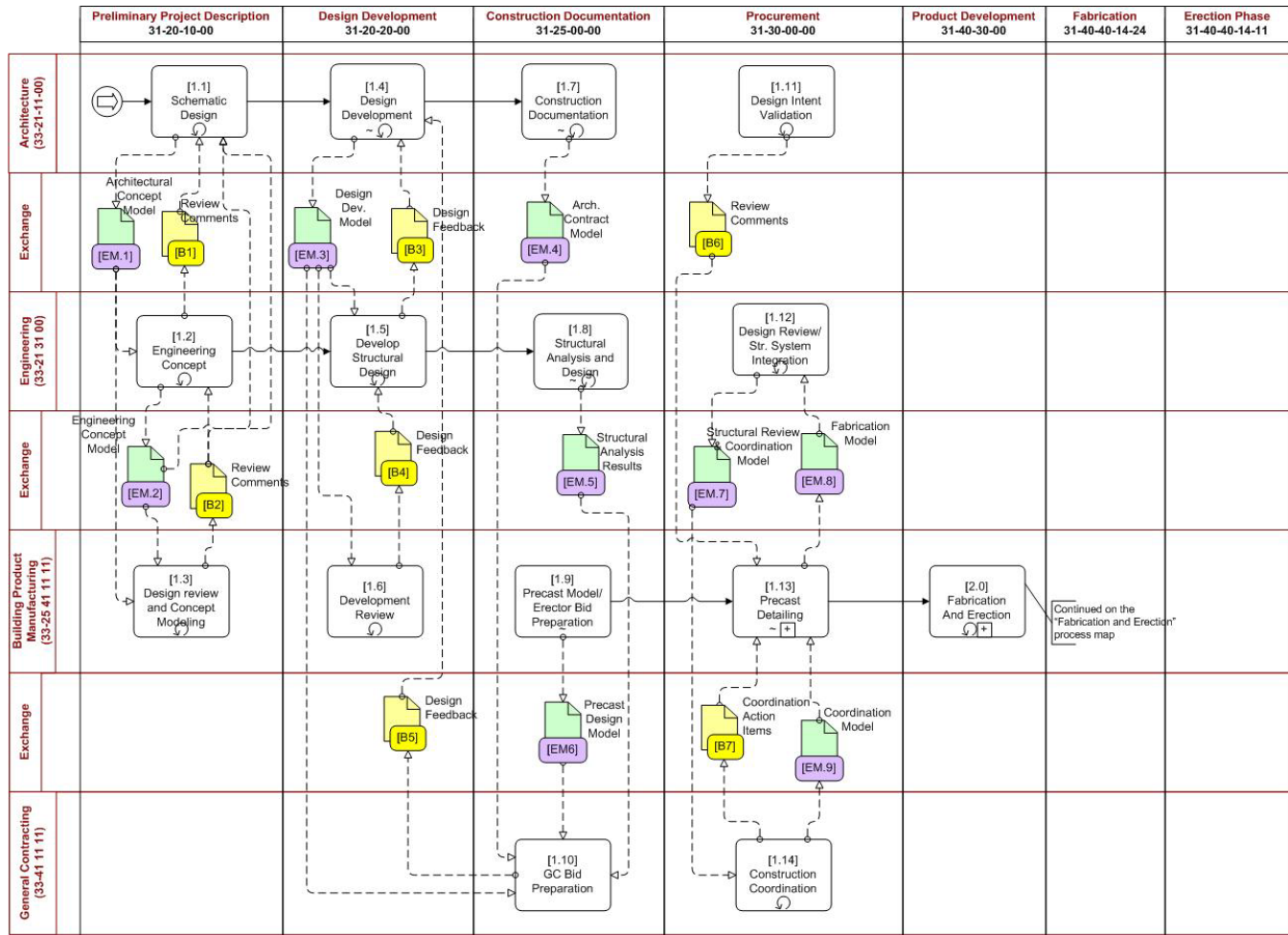


Figure 3: Example process model for Structural Precast

Activities and Use Cases in the Process Map are not meant to describe ALL the Activities within the processes of interest; many will not require data exchanges, and some exchanges may be made between users employing the same software application. Only exchanges between heterogeneous applications are of interest for the NBIMS. Thus the Discipline and Phase swim-lanes are meant to contextualize in a general way the purpose and place in time of a significant exchange.

Table 1: Activity definitions for an IDM Process map-[EM.1] Concept Design of Precast Facade

Type	Activity
Name	Concept Design of Architectural Precast
Omniclass Code	31-20-10-21 Preliminary design stage
Documentation	Architects or designers use an approved or certified BIM authoring application to develop a Building Model that will include non-structural precast façade panels. They define the panel layout, fenestration, and surface patterning. They place structural elements needed to support the precast pieces. They identify elements that are embedded within the precast or are attached to it. The proposed layout may be made available for review in sketch and drawings or as a model

Table 2: Activity definitions for an IDM Process map: [EM.2] Design Review and Concept Modeling

Type	Activity
Name	Design Review and Concurrent Modeling
Omniclass Code	31-20-10-21 Preliminary design stage
Documentation	A precast vendor may be consulted in reviewing candidate layouts of architectural precast facades. The fabricator may comment on manufacturing, shipping, fragility, lifting and erection and other issues that may affect the design. These may be passed as verbal or written comments to the architect.

1.6. Defining the Process Map

The definition of a process map can be undertaken in a fairly systematic manner. The work group should undertake the following tasks, in rough order:

1. Identify the significant Disciplines for which the horizontal row swim-lanes should be designated
2. Validate the project phases in the vertical swim-lanes to ensure coverage of the classes of exchange of interest
3. Identify Activities that have interaction among the primary Disciplines and place them in the project phase where they occur
4. Link Activities that exchange information, especially model information, with a Use case exchange. A Use case may typically consist of one or two model exchanges.

An important decision is made in the layout of Use Cases. If the Use Case has a circular pair (or more) of expected exchanged, these round trip workflows require close coordination, because the second application is updating the data extracted and sent in the first exchange. The issues of round trips are discussed later.

Ideally, the process map is generated by the whole workgroup. In other cases, the workgroup may want to define different Use Case scenarios, for example, to reflect different project delivery modes. The result should be something like the Process Map shown in Figure 3. Each of the Activities should be documented, so that other Workgroups may recollect the intended structure. Examples of the Activity Descriptions are shown in Table 1 & 2.

Last the exchanges should be prioritized. Those exchanges which are most beneficial are marked as critical. If there are many and all are not implemented at the same time, then the order of priority in which they are focused on should be established. These priorities will be useful later, in planning implementation.

1.7. Defining Exchange Requirements and Business Rules

Exchange Descriptions

Exchange Descriptions are a form that identifies the information contents of an exchange. They identify which objects, processes properties, relations and classifications are both relevant to the receiving (importing) application and available in the sending (exporting) application. The Workgroup should easily be able to express a general understanding of what is required, depending on what the exchange is to support, in common English. We use the term 'information items' to refer to the things about which we need to transfer information. These may represent physical objects (such as 'gravity retaining wall', 'precast double tee beam') or abstract ideas (such as 'wind loads', 'surface treatment'). These will be more formally defined in the next section.

Table 3: Exchange Model Descriptions for [A_EM.1] Architectural Concept Model

Project Stage	31-20-10-00 Preliminary Project Description
Exchange Disciplines	(33-21-11-00) Architecture (33-21 31 00) Engineering (33-25 41 11 11) Building Product Manufacturing
Description	Architectural concept model consists of concept layout of precast pieces into simple assemblies, without surface or structural detailing. Building model includes massing models, structural and other grid controls, building program and space layout and use, expected thermal and acoustic functions, if known, It might involve major architectural finishes, structural system selection, structural grid and site analysis.
Related Exchange Models	A_EM.1, P_EM.1, S_EM.1

What we need to accomplish in the full Exchange Description task is to specify these information items and their attributes in sufficient detail that the exchanges will be fully understood regarding their intention by later readers.. They are initially identified in the process maps and are then defined in generic text in the Exchange Descriptions. These are short paragraphs that identify the purpose of an exchange and the general

content, level of detail and expected use of the data in the exchange, defined for later reference. Examples of these Exchange Descriptions are presented in Tables 3, 4 and 5.

Table 4: Exchange Model Descriptions for [P_EM.11] Precast Coordination Model

Project Stage	31-25-00-00 Construction Documentation
Exchange Disciplines	(33-21-11-00) Architecture (33-21 31 00) Engineering (33-25 41 11 11) Building Product Manufacturing
Description	The precast coordination model is an early stage of the precast detail model and is used for coordination of all precast components. It includes detailed model descriptions of all precast structural elements. It is being reviewed by the engineer for structural and logistical consistency.
Related Exchange Models	A_EM.10, P_EM.12, S_EM.9

Table 5: Exchange Model Descriptions for [EM.56] Precast Design Model

Project Stage	31-40-40-14-24 Fabrication Phase
Exchange Disciplines	(33-25 41 11 11) Building Product Manufacturing (33-21-11-00) Architecture
Description	Following EM55, the precast fabricator sends the precast design model to the architect for further review/approval. So in the high level, general information about project site and site buildings are included. Important common categories of information include layout, shape, material types, and information about geometry and materials of finishes, that are covered both in the piece and assembly level. Plus assembly and connection relations of pieces and connections are specified. The piece marks for identification are included. Openings and opening frames are defined. Also, detailed information for some types of products is included. Layout and grid geometry of facades are designated and slab topping thickness, material and surface treatment are determined. The specifications of joints are defined. Nested and assembly relations of both field applied and plant applied connections are specified. Related specifications of other building parts and systems are indicated.
Related Exchange Models	

Exchange Requirements:

Finally, Exchange Requirements are specified in terms of the information items they must carry, fully detailing those outlined in the Exchange Descriptions. They provide clear guidance from the domain experts defining all functional aspects of each exchange. To accomplish this, the Technical Team should develop a specification template that identifies all the functional information of expected relevance in the exchanges. This specification template is meant to provide guidance to the domain experts, allowing them to define the functional aspects of each exchange. This should be based on a review of the following documents and information:

- The applications that are in use in the domain and the capabilities and details of the application's functionality
- The current IFC capabilities dealing with geometry, materials, and features in the domain
- Discussions with the domain experts regarding features, attributes and nomenclature regarding information in exchanges

The possible variations that may be important for a given exchange are identified from the above mentioned documents and information. The variations are organized as an enhanced checklist of possible functional requirements. Often, special concerns need to be taken for geometry, the largest and most complex type of project data in most exchanges. For example, in the domain of precast concrete, the geometry deformation of precast concrete pieces (camber, twisting, deflection, foreshortening), and the accuracy, editability, articulation of features such as connections, blockouts, or surface features and level of detail, need to be identified as possible requirements for exchanges. Embedded parts, including reinforcing and tendons, for connections and seams, and also finishes, especially for architectural panels, need to be recognized if needed for exchanges. Properties and relations between parts may also need to be

specified. Issues of dealing with user-selected subsets of objects and what are minimal subsets for effective exchange should also be considered.

The nesting of physical objects, for example, in case of a precast concrete doubletee (DT) needs to be considered. Within the doubletee are reinforcing meshes, prestressed tendons, and also embedded steel plates for connections. A collection of DTs may also be aggregated into a higher level object (a doubletee slab, which is likely to have its own shape) and have a cast-in-place concrete topping, so that the slab as a whole functions structurally as a diaphragm . In this case, we have the following entities: steel embeds, mesh, tendons, embedded into doubletees, which are aggregated further into a slab. All levels of such aggregations are usually needed in different parts of the design/construction process. Analysis models may be part of some exchanges and the requirements for these include an analytical geometry model, say nodes and members, loading conditions, maximum allowed stresses and deflections, associations between the analytic model and the physical one, and so forth. The full set of information entities needs to be identified if they are required for any of the Use Cases to be developed. Subsets of a broad, inclusive list of entities are easier to review, in comparison to remembering and adding new ones later in the process.

All involved should keep in mind that an application may have certain functionality and data usage that are unique and will not be needed by other applications. In such scenarios only the exchanged information needs to be specified, and not the information carried in a single application.

An exchange model specification consists of a listing of all of the information groups and all of the possible attributes that are needed when making a targeted exchange. We have organized these potential requirements into the following hierarchical levels (also shown in Figure 4):

- Information Groups represent the major classes of objects in a building model such as site, buildings, assemblies, pieces, openings, reinforcing, spaces,

analysis models, connections, processes, etc. These groups should cover the information object classes addressed in the MVD.

- Information Items are specific examples of the members of each information group. They are defined on the assumption that some information items have different attributes from other information items. The information items should cover all aspects of the information group, in broad categories that have similar requirements. As can be seen in the sample exchange specification table shown in Figure 4, the information group ‘Foundations’ has information items ‘Grade Beam’, ‘Pier Cap’, ‘Spread Footing’, etc.
- Attribute Sets are groups of properties that are used to describe an information item. The attributes are grouped in this way because sets occur in identical form across multiple information groups.
- Attributes identify the properties that are needed to fully define the information item.

In application, each exchange model specification must first identify whether a class of object is required, the set of attributes needed if the object is required, and whether each different attribute is required, optional or not needed for its use case. In Figure 4, the attributes are listed in the rows of the table. Each column on the right hand side specifies one Exchange Requirement (i.e. P_EM.1, P_EM.2 and P_EM.3). The cells in that column identify the needs for each information item, at a level that its proper implementation can be defined.



Information Group	Information Items	Attribute Set	Attributes		P_EM.1	P_EM.2	P_EM.3	
Foundations	Grade Beam, Pier Cap, Spread Footing, Slab on Grade, Stem Wall, Retaining Wall, Drilled Pier, Cassion, Pile, Pile Cap	Shape	Geometry	Required?	R	R	R	
				Deformations?	A	A	D	
				Function?	V	F	E	
				Level of	L	M	H	
				Accuracy?	P	P	C	
				Dimensional Tolerance	Required?	O	O	R
		Type	Structural Type (CIP)	Required?	R	R	R	
		Supplier	GC/Contractor/Fabricator	Required?	O	O	O	
		Material	Material type	Required?	R	R	R	
			Quantity	Required?	O	O	R	
	Assembly relations	Part of structural system	Required?			R		
	Nested relations	Contains	Required?			O		
		Contains connection	Required?			O		
	Connection relations	.. to Precast	Required?			O		
		.. to CIP	Required?			O		
		.. to Steel	Required?			O		
	Meta Data	Author, Version, Date	Required?			O		
		Approval Status, Date	Required?			O		

Figure 4: Segment of an Exchange Model Table

Required/Optional/Not needed Property

In the PCI precast project, we classified properties as to whether they are ‘Required’ (R), ‘Optional’ (O) or ‘Not needed’ (blank). In Excel, these can be defined in a pop-up menu for each cell, specifying the allowed alternatives and limiting data entry. ‘Required’ means that if these objects or properties exist in a given building model, the exchange is only valid if the properties have been populated with values and they are included in the exchange. ‘Optional’ means that the exchange is valid whether they are available or not, but indicates that they should be translated if available. In this way, when an exchange model definition is specified for implementation by a software company, a validation check is made whether the exchange file contains the minimum set of objects and their attributes required according to the exchange model specification.

Business Rules:

The level of detail in the provided and exchanged models for each information unit can vary based on the project stage, purpose of model exchange, model recipient and local practices. Further, different project delivery methods impose changes in roles and

responsibilities of project parties, which considerably change project deliverables at each stage for each discipline involved in the project. Hence, a finer level of adjustments in exchange objects needs to be provided to make them applicable in different exchange models and localities. Business rules identify these restrictions on the data structures and/or on the attribute values that may be applied in some of the Use Case contexts considered in the MVD.

Objects are often grouped in different ways, for example, in the context of an erection sequence, fabrication runs, purchasing, etc. While often these can be addressed within a single application, they occasionally are broadly applied. In steel and precast concrete, for example, the piecemark identifier groups similar pieces because they are made in the same production run and may be interchangeable in erection. The drawing they are produced from is called a piece ticket. They are potentially different from type and instance, in that they may be the same from a production standpoint, but are modified slightly for a particular project location. These piecemarks must be managed like GUIDs, serving as an important identifier in made-to-order products.

Approvals: Some exchanges are for the purpose of review, revision or approval. For effective processing and review, the parts of the design being reviewed need to be grouped, then assessed and acted upon. The business practices for such actions can be quicker, and more reliable because of 3D geometry and management of their associated properties. All of this should be identified in the business rules, as the Workgroup begins to get a handle on these issues.

1.8. Scoping Issues:

Given the range of possible information, the task of defining *exchange* requirements can be quite large. Here is a good time to determine what subset *of exchanges* will be implemented first, second and so forth. Quick, visible benefits are to everyone's advantage. Too ambitious an initial effort may lose the motivation needed to complete it.

1.9. Workflow Details

Round trips: If two different applications can both modify a project model, then they may iterate exchanges, passing the model back and forth. An example might be the architect and fabricator, working within an Integrated Project Delivery (IPD) contract. Again, we use precast panels for our example. The architect generates an initial panel layout and patterning for a mid-rise building. The architect does not know production or erection constraints or options, or detailing issues. Thus the initial design does not reflect these issues and they will be proposed by the precast fabricator as modifications. Multiple iterations might be required.

In such a set of exchanges, the full set of entities could be passed and after each iteration users start from this full set of entities for review. A better method however, is to only exchange the changed objects, reducing the scope of issues to be reviewed and also file size. In our example, the architect may send a full façade (or the whole set of building façades). What he/she gets back are those that the precast fabricator has proposed be modified. Some are approved and others possibly rejected, for which the architect proposes other alternatives. The new variations are sent back to the precaster for review, and so forth.

The issue of round trips are not explicitly a model view issue, but rather how applications support exchange transactions. Round trips require an application to not only manage the data it has produced, but to support the data generated at the other end of the exchange, including attributes. The management and merging of round trip data

are points of discussion to have with BIM tool application developers. Round trips are hard to support and not widely used in IFC exchanges, but they really facilitate how people can work together. Most people are familiar with sharing development of a marketing presentation, or similar report. This same level of smooth exchange is supported by IFC, but requires effective transaction support to be associated with Model View Definition.

1.10. The Information Delivery Manual

The product of the above work forms the basis of the Information Delivery Manual (IDM) for a given NBIMS project. As explained previously, forming a Workgroup, developing a Process Map, defining the set of Use Case Exchanges being addressed by the Workgroup, describing the Activities involved and most importantly, the Exchange Requirements are included in an IDM. It captures the user needs and specification of the exchanges in a form that can be translated into technical exchange specifications, called a Model View Definition (MVD), which is the next stage of this effort. The completed IDM document set must be voted on and approved by the Workgroup. This is done using the consensus voting process established for NBIMS.

1.11. Generic BIM User Guide

At the end of this phase, the buildingSMART NBIMS procedure also recommends development of an outline draft of an Exchange User Guide for the Use Case exchanges. This is practical if the Use Cases are singular or few in number and their use context is well defined. If they are for a whole domain, each application and context of use will have its own processing and context requirements. In these cases, the user guide should be outlined to define what topics are to be addressed in the user guide, to provide guidance for software implementation groups.

The context and intended use of each Use Case should be documented. Also, the entities that to be included and their level of development/detail should be defined. These general aspects will be elaborated later, when the next phases are completed.

1.12. IDM Submission of buildingSMART North America

The approved IDM report is submitted to the buildingSMART Project Committee with the internal approval noted. The buildingSMART Project Committee is expected to review it and do a functional mapping with other existing IDMs to check for overlaps and potential duplication or inconsistencies. If these are found, they are communicated back to the workgroup, so that coordination can take place. It should also be submitted for inclusion on the IDM/MVD web site (www.blis-project.org/IAI-MVD) as the requirements definition for an MVD development Project.

2. PHASE 2: DESIGN

2.1. Translating Exchange Requirements into an MVD

Phase 2 Overview: The Design Phase of NBIMS creates the Model View Definition (MVD) binding and specification needed to implement the exchanges defined in the IDM. This task is largely a technical one carried out by the Technical Advisory team.

2.2. Modularization in Model Views

This process has evolved and been refined over the last few years. It was quickly noted that the contents in different model views, but within similar domains, were often replicated; contents in different model views, but within similar domains, were often replicated. So they developed and used the notion of data exchange modules that could be reused in many MVDs. The reusable modules are called “Concepts”.

The Concept based approach has been developed jointly by European and North American groups and is being widely adopted because of its improved support for software implementation. Concepts represent semantic units that map the Exchange Concepts (developed in the ERM) to an information model schema (most often IFC). The Concept-based approach also allows MVD domain-specific groups such as ours to re-use Concepts that have already been developed by others. This reuse includes the online implementer documentation and any testing that has already been developed for the concept. This approach also allows software companies to implement a Concept once, and then re-use it in many MVD exchanges.

The software companies we consulted throughout the Precast MVD development process are familiar with the IFC Solutions Factory website and its use. It is becoming accepted as the standard implementation approach for MVDs, including those in Europe, those sponsored by GSA, and other North American initiatives. We recommend using the Concept based approach and the associated IFC Solutions Factory website.

A Note: When defining new Concepts, there is no precise or rigorous method for partitioning a model into Concepts. At this time, different groups define them in somewhat different ways. Some generate a concept for each attribute; others define concepts that incorporate attribute sets, or even represent full information items. An important requirement, which was identified during the current model view work, is the need to avoid redundancy and rework in terms of development and testing of Concepts. Hence, concepts should be generated following strict/formal rules so that they are testable and standalone. For new MVD development, the requirements should be specified in a modular form using such concepts. Redevelopment of Concept hierarchies, and the subsequent reworking of MVD definitions is expensive and time consuming. Where possible, this should be avoided. From a semantic point of view, there should be no broken links or references and semantic relationships among terms should be explicitly defined. Your judgment in these issues will be required. .

2.3. Exchange Requirement Model

Development of an Exchange Requirements Model (ERM) is the first step in translating the end user requirements for data exchange defined in the IDM into data exchanges in software products. An ERM's purpose is to define the high level structure for the to-be-developed Concept structure. It is essentially a high-level graph of objects which composes the information defined in the Exchange Requirements in the IDM into diagrams showing the extent and relationships between the data for each high level concept (e.g. Wall, Door, Window). Generally speaking, there should be a 1:1 correspondence between the items defined in the tabular Exchange Requirements and the items in an ERM. An example is shown in figure Five

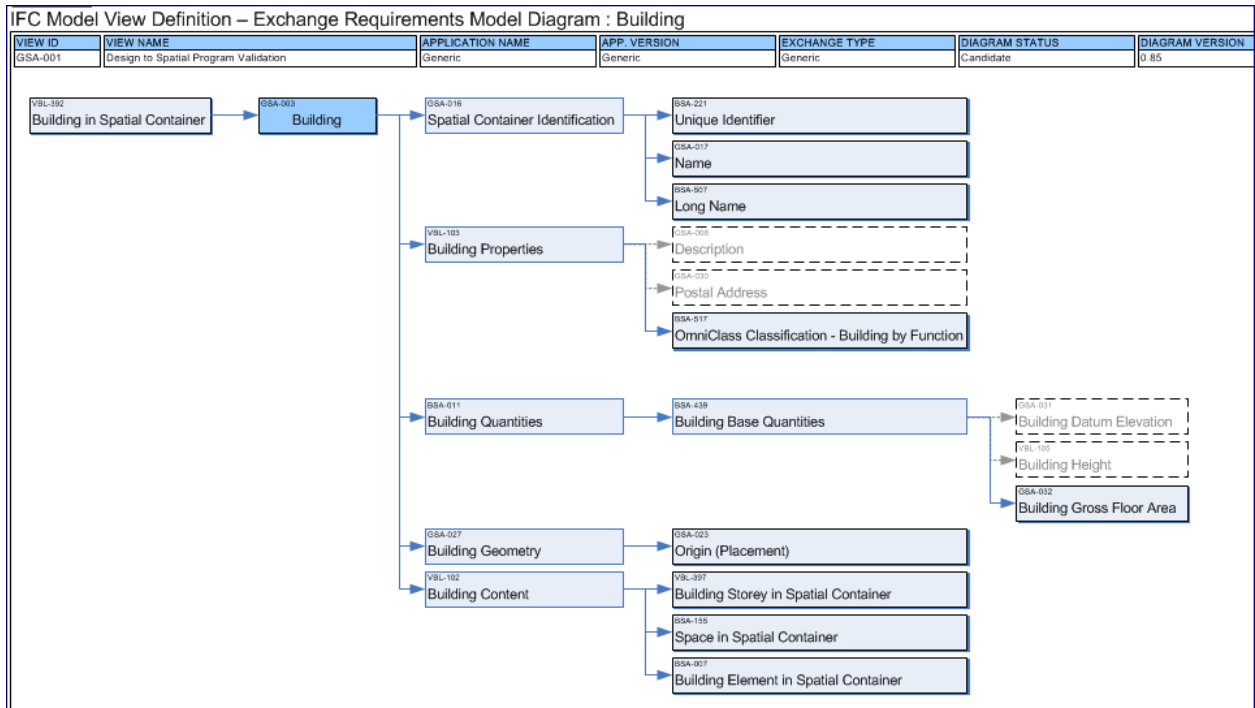


Figure 5: ERM Diagram for Building – Design to Spatial Program Validation IDM/MVD

2.4. Model View Definition

The development of Concepts and structuring them to formulate a Model View Definition (MVD) is primarily technical work; it involves translating the functional intent defined in the IDM documents, based on good knowledge of the IFC or other targeted information model schema.

The initial step is the mapping of Exchange Concepts, as defined in the ERM, to MVD Concepts. MVD Concepts define a representation for the data defined in the driving top level Exchange Concepts. These Concepts are structured according to the selected information model schema and enable exchange of the subject data between two software applications. MVD Concepts are structured hierarchically. The Leaf nodes in this hierarchy are called **Static Concepts**. They are called Static because their definition does not change. That is: they are exactly the same in all MVDs that include them. This means implementation in software products will also be the same, leading to efficiency and reuse in software implementations. Static Concepts are then

aggregated into higher level **Adapter Concepts**, allowing the higher level Concepts to be re-used as a group where needed. Finally, at the top of the hierarchy of concepts is a **Variable Concept**. Variable Concepts are so named because their definition (through the tree of Adapter and Static Concepts in the hierarchy) will vary between MVDs. That is: the data about (e.g.) a Wall will incorporate different base and Static Concepts (in response to the IDMs that define exchange requirements for those MVDs). Each diagram in the MVD is focused on a single Variable Concept. Examples include: Wall, Door, and Window.

The scope of an MVD can vary greatly, because an MVD can address the exchange requirements of one or many related IDMs. If the MVD addresses a broad AEC domain, such as precast, reinforced concrete or steel structures, the MVD Concepts and associated binding to an information model will be numerous and span a broad range of exchange requirements. For the precast concrete example above, this will include all the Static Concepts needed for exchange regarding precast concrete design, production planning and fabrication. These are then selectively grouped to address the information coverage in multiple exchanges defined in multiple Exchange Requirements. Almost certainly, this will be a mixture of new Concepts and also the re-use of existing ones listed on IFC Solutions Factory. As the number of NBIMS projects grows, the number of available 'off-the-shelf' Concepts will also grow. Eventually, a new Use Case may require only a few or even no new Static Concepts, being composed entirely out of existing ones. That is the goal of the collaborative Solutions Factory Site – to develop a rich set of MVD concepts that are shared across many MVDs, ensuring efficiency, consistency, and predictable interoperability experiences for the AEC industry.

On the other hand, if the MVD addresses one or a small number of Exchange Models, then the Exchange Requirements can be directly defined in terms of largely Concepts, without needing to define first the intermediate ones in the domain.

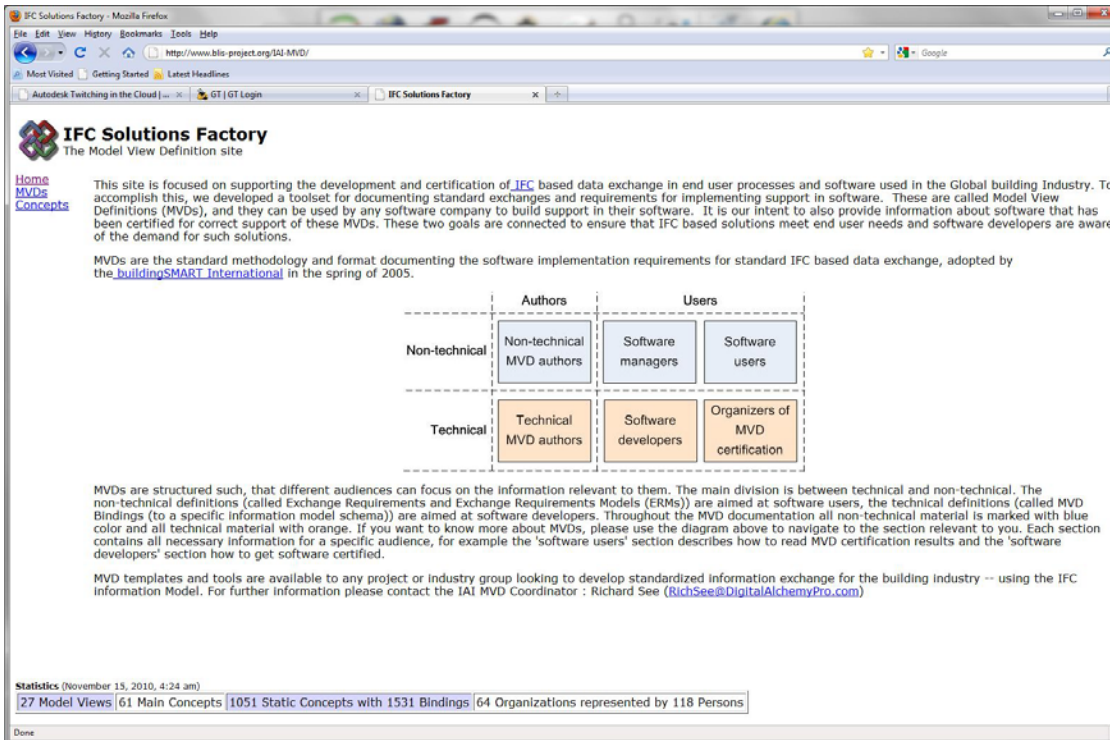


Figure 6: Front page of the IFC Solutions Factory.

2.5. A Guide to IFC Solutions Factory

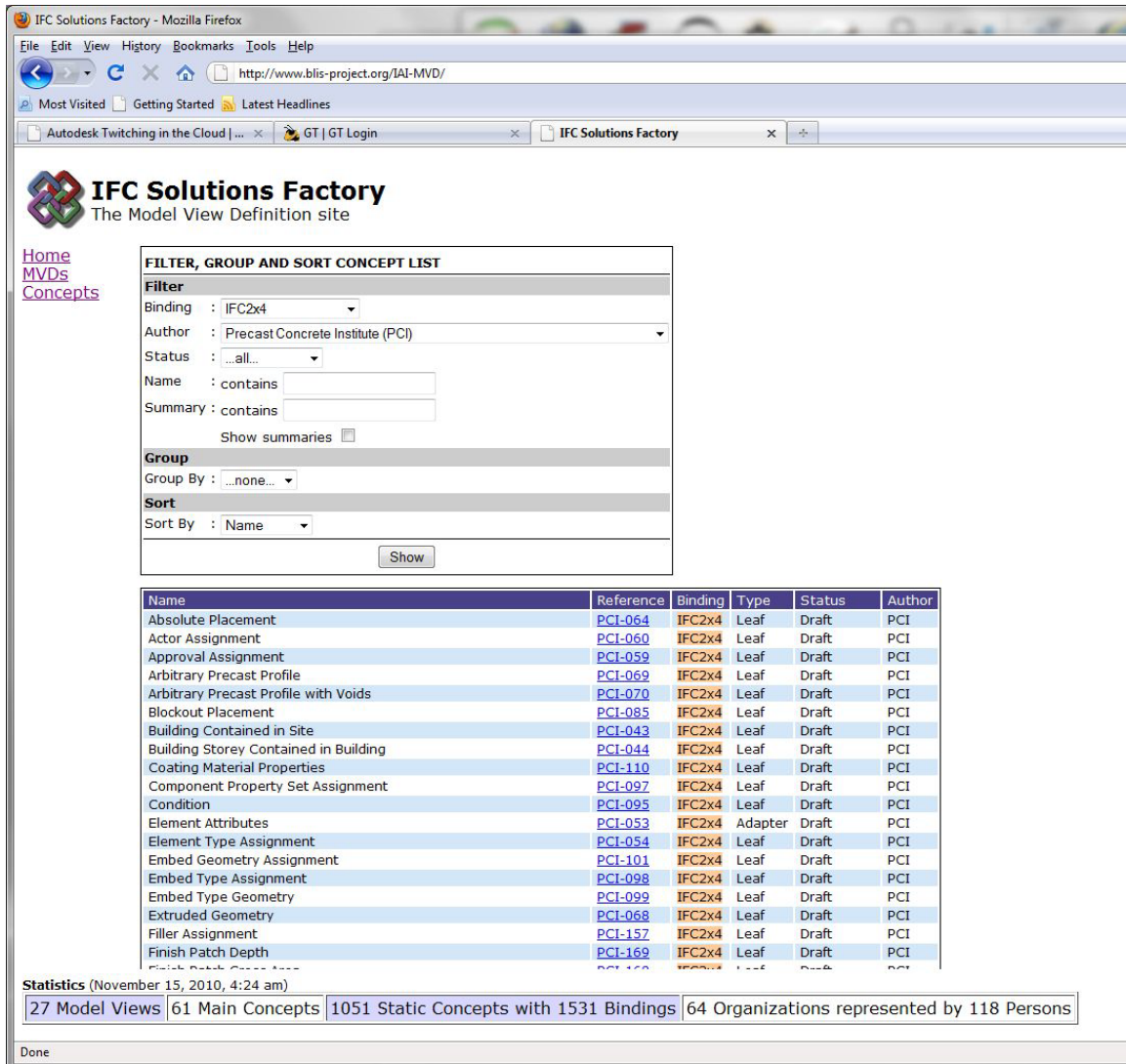
Here is a walk-through guide to the structure of information in the IFC Solutions Factory. The IFC Solutions Factory website is at: <http://www.blis-project.org/IAI-MVD/>. It is an open and public international website for integrating IFC Model View Definitions. It is a carefully structured and hyperlinked website that will support vendors in their implementation. It requires registration of both the MVD as a project and registration of each individual user participating in the project. The main site is shown in Figure 6.

After accessing the website, by clicking the MVD link in the upper left corner, the different MVDs are listed, by sponsoring organization and reference ID. It shows MVDs sponsored by GSA, ATC, PCI, CRC (Australian Construction Research Council), ICC (International Code Council), and others. The listing, as of late 2010, is shown in Table 6.



Table 6: List of MVD development activities making use of IFC Solutions Factory

1.1 Exchange Model	1.2 Organization
1. Architectural design to circulation/security/analysis	US General Services Administration
2. Architectural design to landscape design	CRC for construction innovation
3. Architectural design to quantity take-off – level 1,2,3	Virtual Building Laboratory; German Speaking Ch.
4. Architectural design to spatial program validation	US General Services Administration
5. Architectural design to struct. design and to structural analysis	Virtual Building Laboratory @ TUT
6. Architectural design to thermal insulation	Virtual Building Laboratory @ TUT
7. Architectural programming to architectural design	BuildingSmart International
8. Basic handover to facility management	German Speaking Chapter
9. Concept design BIM 2010	US General Services Administration
10.Design to code compliance checking	International Code Council
11.Design to energy performance analysis	Building Smart Alliance, North America
12.Design to quantity take-off	Building Smart Alliance, North America
13.Extended coordination view	IAI Implementers Support Group
14.Extensibility	Virtual Building Laboratory @ TUT
15.Indoor climate simulation to HVAC design	Helsinki University of Technology – HVAC Lab
16.Landscape design to road design	CRC for construction innovation
17.Precast Concrete Exchanges	Precast Concrete Institute
18.Road design to landscape design	CRC for construction innovation
19.Space requirements and targets to thermal insulation	Helsinki University of Technology – HVAC Lab
20.Structural design to structural detailing	Applied Technology Council



IFC Solutions Factory
 The Model View Definition site

[Home](#)
[MVDs](#)
[Concepts](#)

FILTER, GROUP AND SORT CONCEPT LIST

Filter
 Binding : IFC2x4
 Author : Precast Concrete Institute (PCI)
 Status : ...all...
 Name : contains
 Summary : contains
 Show summaries

Group
 Group By : ...none...

Sort
 Sort By : Name

Show

Name	Reference	Binding	Type	Status	Author
Absolute Placement	PCI-064	IFC2x4	Leaf	Draft	PCI
Actor Assignment	PCI-060	IFC2x4	Leaf	Draft	PCI
Approval Assignment	PCI-059	IFC2x4	Leaf	Draft	PCI
Arbitrary Precast Profile	PCI-069	IFC2x4	Leaf	Draft	PCI
Arbitrary Precast Profile with Voids	PCI-070	IFC2x4	Leaf	Draft	PCI
Blockout Placement	PCI-085	IFC2x4	Leaf	Draft	PCI
Building Contained in Site	PCI-043	IFC2x4	Leaf	Draft	PCI
Building Storey Contained in Building	PCI-044	IFC2x4	Leaf	Draft	PCI
Coating Material Properties	PCI-110	IFC2x4	Leaf	Draft	PCI
Component Property Set Assignment	PCI-097	IFC2x4	Leaf	Draft	PCI
Condition	PCI-095	IFC2x4	Leaf	Draft	PCI
Element Attributes	PCI-053	IFC2x4	Adapter	Draft	PCI
Element Type Assignment	PCI-054	IFC2x4	Leaf	Draft	PCI
Embed Geometry Assignment	PCI-101	IFC2x4	Leaf	Draft	PCI
Embed Type Assignment	PCI-098	IFC2x4	Leaf	Draft	PCI
Embed Type Geometry	PCI-099	IFC2x4	Leaf	Draft	PCI
Extruded Geometry	PCI-068	IFC2x4	Leaf	Draft	PCI
Filler Assignment	PCI-157	IFC2x4	Leaf	Draft	PCI
Finish Patch Depth	PCI-169	IFC2x4	Leaf	Draft	PCI

Statistics (November 15, 2010, 4:24 am)
 27 Model Views | 61 Main Concepts | 1051 Static Concepts with 1531 Bindings | 64 Organizations represented by 118 Persons

Figure 7: Concept Filter menu and example listing of Concepts.

The list of Concept can be accessed by clicking on the Concepts button in the upper left menu in IFC Solutions Factory webpage. A filter for different types of Concepts is presented, shown in the top of Figure 7. It allows selection of Concepts by:

- IFC binding - which version of IFC are the Concepts based on
- Author - group that defined the Concepts
- Status:



- all - all Concepts
 - Placeholder - named but not filled in,
 - Draft - complete but not implemented
 - Proposal - in review for implementation by software companies
 - Candidate - implemented and awaiting certification
 - Official - implemented and certified.
 - Deprecated - earlier version Concept, no longer used.
-
- Partial name - string within the Concept name
 - Partial summary - string within Concept summary

By selecting an authoring group, the Concepts authored by the group are listed. An example is shown at the bottom of Figure 7.

By clicking on the left arrow of a listed MVD, the information about the selected MVD is expanded. In the expanded tableau, clicking on “Exchange Requirements” the IDM report is opened for review. This is the document providing the functional specifications for the information provided in the MVD. Clicking on “Definition - Overview” the scope of the MVD and its objectives are defined in a template set up for this purpose. The bindings are recorded as being for IFC Release 2.x3 or 2.x4. By clicking on “Bindings – Overview” the range of exchanges are identified. By clicking on “Binding – Diagram”, the Concepts that are in the MVD are laid out in a table, as shown in Figure 8. The left side brighter orange colored Concepts are the Variable ones.



IFC Model View Definition Diagram : [PCI-001] Precast Concrete

APPLICATION NAME	APP. VERSION	EXCHANGE TYPE	DIAGRAM AUTOCAD
Generic		Generic	Rafael Saez, Chuck Eastman and Ivan Panathier
000-IPCSA Building	000-411 Building Attributes	000-000 Property Set	000-000 Absolute Placement
000-IPCSA Building Storey	000-412 Building Storey Attributes	000-111 Reinforcing Bar Attributes	000-000 Actor Assignment
000-IPCSA Engineered Mesh	000-000 Element Attributes	000-000 Reinforcing Element Property Set Assignment	000-410 Aggregates: Structural Analysis Models
000-IPCSA Grids	000-100 Embed Geometry Assignment	000-100 Reinforcing Unit Attributes	000-100 Applied Load
000-IPCSA Non-precast Element	000-000 Generic Aggregation	000-000 Shape Representation	000-000 Approval Assignment
000-IPCSA Non-precast Element Type	000-000 Generic Assignments	000-010 Site Attributes	000-000 Arbitrary Precast Profile
000-IPCSA Precast Blockout	000-000 Generic Associations	000-010 System Serves Building Space	000-000 Arbitrary Precast Profile with Voids
000-IPCSA Precast Embed Type	000-000 Generic Geometric Representation	000-010 System Serves Building Storey	000-000 Area
000-IPCSA Precast Embeds	000-000 Generic Material Association	000-110 Tendon Attributes	000-000 Assigns Other Building System
000-IPCSA Precast End-to-edge Connection	000-000 Generic Object Placement	000-110 Tendon Debonding Attributes	000-000 Assigns Precast Facade System
000-IPCSA Precast End-to-end Connection	000-000 Generic Shape Representation	000-100 Transverse Bar Attributes	000-000 Assigns Precast Structural System
000-IPCSA Precast Joint	000-000 Grid Attributes		000-000 Blockout Placement
000-IPCSA Precast Joint Type	000-000 Longitudinal Bar Attributes		000-000 Building Contained in Site
000-IPCSA Precast Piece	000-000 Mechanical Attributes		000-010 Building Name
000-IPCSA Precast Piece Type	000-000 Piece Type Geometry Assignment		000-000 Building Storey Contained in Building
000-IPCSA Precast Projection	000-000 Precast Blockout Attributes		000-010 Building Storey Name
000-IPCSA Precast Seam Connection	000-000 Precast Component Assignment		000-110 Coating Material Properties
000-IPCSA Precast Slab	000-100 Precast Connection Attributes		000-000 Component Property Set Assignment
000-IPCSA Project	000-100 Precast Connection Element Assignment		000-000 Condition
000-IPCSA Rebar	000-100 Precast Embed Attributes		000-110 Description
000-IPCSA Rebar Cage	000-100 Precast Embed Type Attributes		000-000 Element Quantity
000-IPCSA Reinforcement Element Aggregation	000-000 Precast Finish Assignment		000-000 Element Type Assignment
000-IPCSA Site	000-000 Precast Piece Material Association		000-000 Embed Type Assignment
000-IPCSA Standard Mesh	000-100 Precast Piece Structural Member Assignment		000-000 Embed Type Geometry
000-IPCSA Tendon	000-000 Precast Piece Type Attributes		000-000 Extruded Geometry
	000-100 Precast Point Connection Structural Model		000-000 Generic Brep Shape Geometry
	000-100 Precast Projection Attributes		000-000 Grid Axis Assignment
	000-000 Precast Property Set Assignment		000-000 Grid Name
	000-100 Precast Seam Connection Structural Model		000-000 Grid Representation
	000-000 Production Attributes		000-000 Grid Spatial Structure Containment
	000-000 Project Attributes		000-110 GUID
	000-000 Project Units		000-000 Imperial Project Units
			000-000 Material Name
			000-100 Mesh Length
			000-100 Mesh Width
			000-000 Metric Project Units
			000-100 Name
			000-010 Owner and Status Information
			000-000 Placement Relative to Grid
			000-000 Precast Blockout Assignment
			000-100 Precast Connection Component Assignment
			000-000 Precast Design Criteria
			000-000 Precast Embed Assignment
			000-100 Precast End-to-edge Connection Geometry
			000-100 Precast End-to-end Connection Geometry
			000-000 Precast Fabrication Attributes
			000-000 Precast Finish
			000-000 Precast General Attributes
			000-100 Precast Joint Attributes
			000-100 Precast Joint Element Assignment
			000-100 Precast Joint Location
			000-100 Precast Joint Type Assignment
			000-100 Precast Joint Type Attributes
			000-100 Precast Joint Type Profiling Geometry
			000-000 Precast Piece Containment
			000-000 Precast Piece Mark
			000-100 Precast Point Connection Reaction
			000-100 Precast Projection Assignment
			000-000 Precast Rebar Assignment
			000-100 Precast Seam Connection Location
			000-100 Precast Seam Connection Reaction
			000-000 Precast Slab Aggregation
			000-000 Production Control Number
			000-000 Project Name
			000-010 Project Phase
			000-010 Pset_Precast_Building_Design_Criteria
			000-100 Rebar Association to Engineered Mesh
			000-100 Rebar Association to Standard Mesh
			000-110 Rebar Bending Attributes
			000-000 Rebar Extruded Shape Geometry
			000-000 Reinforcing Element Aggregation Association to Rebar Cage
			000-000 Reinforcing Element Association to Reinforcing Element Aggregation
			000-100 Reinforcing Mesh Attributes
			000-000 Reinforcing Unit Association to Piece
			000-000 Relative Placement
			000-000 Site Contained in Project
			000-000 Site Curve 2D
			000-010 Site Name
			000-000 Space Contained in Building
			000-000 Space Contained in Building Storey
			000-100 Steel Material Properties
			000-000 Supplier
			000-000 Surface Treatments
			000-000 System Piece Aggregation
			000-100 Tendon Anchorage Slip
			000-100 Tendon Debonding Length
			000-100 Tendon Debonding Material
			000-100 Tendon Debonding Relative Placement
			000-000 Tendon Extruded Shape Geometry
			000-100 Tendon Friction Coefficient
			000-100 Tendon Min Curvature Radius
			000-100 Tendon Prestress
			000-100 Tendon Tension Force
			000-000 Volume



Figure 8: The list of Variable Concepts (left row) and set of Static (leaf) Concepts with IFC bindings for the precast Concrete Domain Model Views.

By clicking on any of the Variable Concept boxes in this table, the Adapter Concepts are opened and described. The Variable Concept for Precast Piece is shown in Figure 8. By clicking on a Static (leaf) Concept, the implementation of that Concept is defined in a right side overlapping window, as shown in Figure 9.

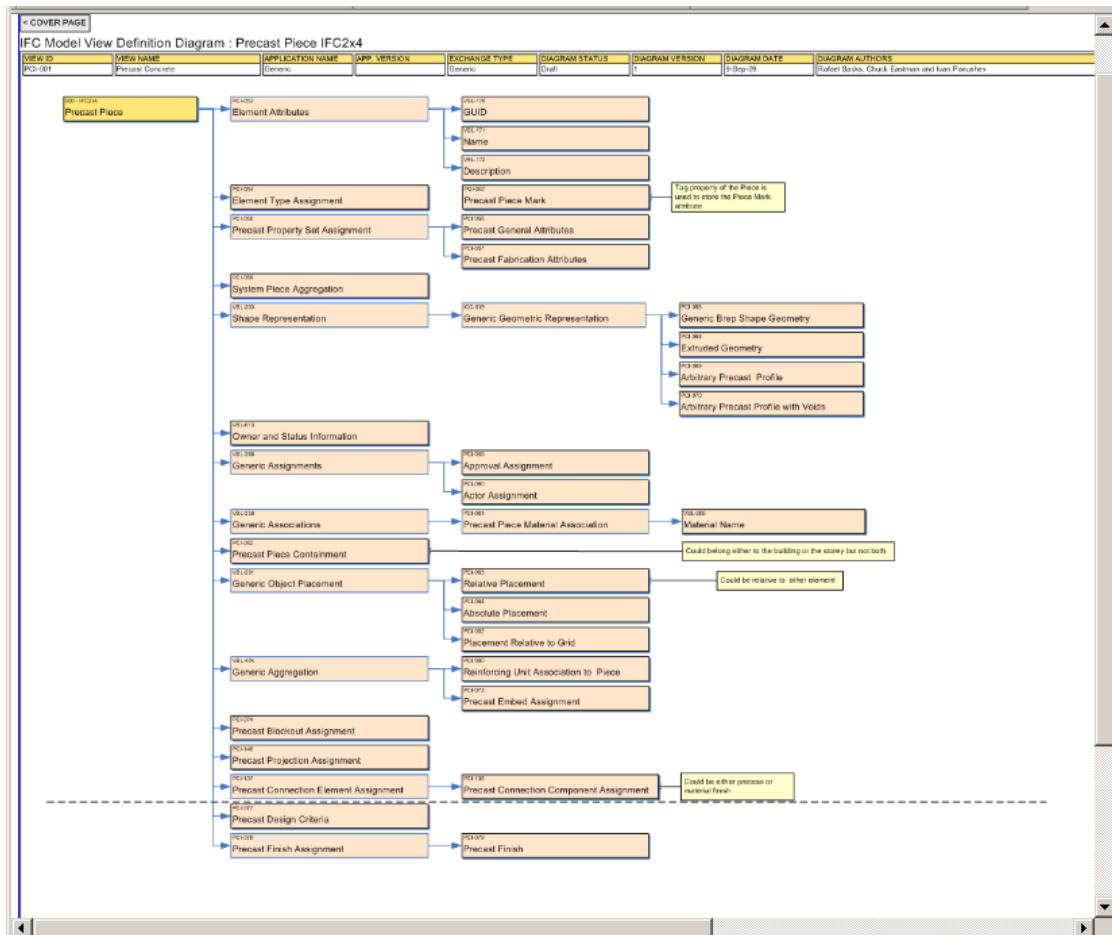
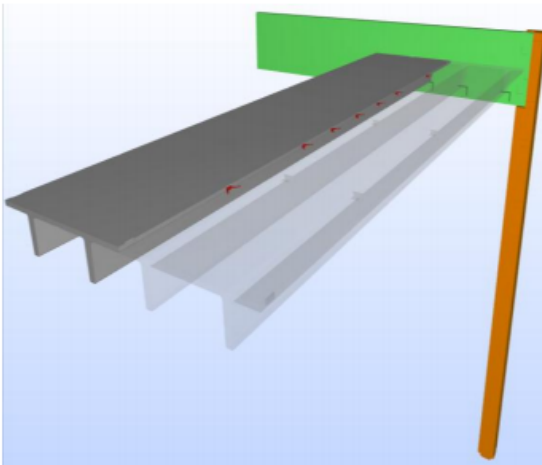


Figure 9: A Variable Concept, showing the more detailed Adapter Concepts for the Precast Piece Variable Concept.

An example for Precast Connection Component Assignment is shown in Figure 10. The static binding diagrams identify the IFC Entities and their references to each other for different uses. They are still abstracted, in that these diagrams omit IFC Types, including Enumerated Types and Select Connection Types, all important low-level Entities in IFC. However, they are easily resolved in implementation. At the bottom part of each

Binding diagram page is a list of the attributes for each entity. They indicate the assignments and any restrictions that might apply in the implementation (also called business rules). These are also used to resolve any ambiguities in the range of attribute assignments or types. Last, a segment of the IFC Part-21 instance file is provided for

IFC Release Specific Concept Description (IFC 2x4)					
Precast Connection Component Assignment					
Reference	PCI-136	Version	1.0	Status	Draft
Relationships					
History	v.1.0 8-Aug-09				
Authors	Rafael Sacks				
Document Owner	Precast/Prestressed Concrete Institute (cvsacks@technion.ac.il)				
Description					
Defines the relationship between a precast connection and its component parts, which may be any of:					
<ul style="list-style-type: none"> - Plant-applied connection components (discrete accessories - plates, anchors, etc.) - Field-applied connection components (discrete accessories - plates, shims, bolts, etc.) - Precast concrete projections, such as a corbel or shelf added to a column or a spandrel - Reinforcement bars - Blockouts 					
Usage in view definition diagram					
-FC20	Precast End-to-end Connection	→	FC136	Precast Connection Component Assignment	
-FC20	Precast End-to-edge Connection	→	FC136	Precast Connection Component Assignment	
-FC20	Precast Seam Connection	→	FC136	Precast Connection Component Assignment	
Instantiation diagram					



```

IfcRelConnectsWithRealizingElements
+ GlobalId
+ OwnerHistory >
Name
Description
ConnectionGeometry >
+ RelatingElement >
+ RelatedElement >
+ RealizingElements >
ConnectionType
                    
```

```

IfcDiscreteAccessory
+ GlobalId
+ OwnerHistory >
Name
Description
ObjectPlacement >
Representation >
Tag
                    
```

```

IfcReinforcingBar
+ GlobalId
+ OwnerHistory >
Name
Description
ObjectType
ObjectPlacement >
Representation >
Tag
SteelGrade
+ NominalDiameter
+ CrossSectionArea
BarLength
+ BarRole
BarSurface
                    
```

```

IfcProjectionElement
+ GlobalId
+ OwnerHistory >
Name
Description
ObjectPlacement >
Representation >
Tag
                    
```

```

IfcOpeningElement
+ GlobalId
+ OwnerHistory >
Name
Description
ObjectType
ObjectPlacement >
Representation >
Tag
                    
```

Implementation agreements

Attribute	Implementation agreements
RealizingElements	Must be IfcDiscreteAccessory, IfcReinforcingBar, IfcProjectionElement (a projecting feature such as a corbel) or IfcOpeningElement (a blockout, such as a pocket).

Note: Realizing elements of type reinforcing bar or discrete accessory may be embedded in one of the precast pieces that is part of the connection, or they may be delivered to the site as field hardware. In the former case, the element must also be associated directly with the building element in which it is embedded using an aggregation relationship, in addition to its relationship to the connection as defined here. For details, see "Precast Embed Assignment" concept.

Example Part 21 File for End to End Connection

```

#21= IFCARTESIANPOINT(0,0,0);
#25= IFCDIRECTION(1,0,0);
#29= IFCDIRECTION(0,1,0);
#33= IFCDIRECTION(0,0,1);
#37= IFCANIS2PLACEMENT3D(#21,#33,#25);
...
#115= IFCRECTANGLEPROFILEDEF(AREA,400*400,#112,400,400);
#116= IFCDIRECTION(-1,0,0);
#120= IFCDIRECTION(0,0,-1);
#124= IFCARTESIANPOINT(0,0,4000);
#128= IFCANIS2PLACEMENT3D(#124,#120,#116);
#131= IFCINTRUDEDAREAASOLID(#115,#128,#33,4000);
#134= IFCSTYLEDELETEDEM(#131,#102,Name);
#138= IFCSHAPEREPRESENTATION(#40,Body/SweptSolid,/#131);
#144= IFCPRODUCTDEFINITIONSHAPE,/#138);
                    
```

Figure 10: An IFC binding, defined as a Static Concept.

most Concepts to provide a specific example of how they are to be defined, in the format of an IFC text file (used in most exchanges).

This structure is readily available to software companies, for PCI-related work, or to other projects needing to exchange the same or similar information.

The Technical team of the Precast Concrete NBIMS project invested multiple person years to define the broad set of Concepts needed to address the range of information about precast concrete over its design/fabrication/erection lifecycle. It involves learning the IFC schema and its conventions intimately. The documentation dealing with IFC is improving significantly with 2x4, which should facilitate understanding of good use of the entities for composing new Concepts.

2.6. Definition of Concepts for Uploading

The major work undertaking the MVD development involves:

1. Defining the IFC bindings in Static Concepts
2. The aggregation of Static Concepts and Variable Adapter Concepts for re-use
3. Uploading the concepts for public (and software company) access.

The above steps are described in order.

Defining the IFC Bindings in Static Concepts

It is suggested that Concepts be defined from the bottom up, starting from the IFC bindings, so that it is worked out how the Concepts will interface and not overlap when they are aggregated. This step requires full and constant access to the IFC documentation. In order to prepare the MVD information for uploading, the following three sets of materials are needed:

- Template sheets in Word format for defining IFC bindings, as shown in Figure 9. A Version is provided in [Appendix A](#).

- The Visio Shapes of IFC Entities for the current IFC Release. These will soon be downloadable from the IFC Solutions Factory website¹.
- The Visio MVD shapes, for links and INVERSE references, as shown in Figure 9. These are accessed the same as for the IFC Shapes.

In undertaking these mappings from functional requirements in the IDM to Concepts, we proceeded by grouping similar functional requirements together and assigning them to different members of the Technical Committee. A particular functional requirement from the IDM could be mapped and implemented with multiple alternative structures within IFC. These require review of other similar Concepts and occasionally, advice from IFC implementation advisors. In parallel, we composed and aggregated these Static Concepts into higher level Adapter and Variable Concepts that were effective for our own re-use. Each individually developed Concept was reviewed by the full Technical team for correctness and consistency.

The following are step-by-step instructions for filling in the Binding template page (an example of the page is shown in Appendix A).

First, the fields in the template page for bindings should be filled in. These are data fields and have associated macros. They must be assigned as values to fields within Word, using the 'Insert' pull-down menu and 'Field' operation. The assignments should be as follows:

- *<IFC Release Field>* -- assigned the IFC Release that the Concept is using.
- *<Title field>* -- assigned the name of the concept that uniquely distinguishes it from other Concepts. This name is used to reference this Static Concept for

¹ Currently, one can obtain them by writing directly to the IFC Solutions Factory site manager (as of writing, RichSee@DigitalAlchemyPro.com).

incorporation into higher level Adapter and Variable Concepts, and for global indexing in IFC Solutions Factory.

- *<Reference field>*, *<Version field>* and *<Status field>* -- all managed centrally by the uploading service.
- *<Company field>* -- also a field, with company prefix and number automatically assigned.

The other fields are manually defined, using simple text. These should have the following contents:

- *Relationship*: should indicate how this Concept is related to others, sometimes left blank
- *History*: The version date should be provided here; revisions should be noted and dated
- *Authors*: The Author of the Concept and their web access information should be provided, in case there are questions regarding the Concept.

The Usage in View diagrams should show which Adapter Concepts use this Static Concept. An example is shown in Figure 9. These are hyperlinked as pdf files.

The Visio binding diagram, showing the shapes that represent IFC Entities and the structure that connects them, should be pasted from Visio into the template. It should not be a jpeg image, but a hyperlinked Visio file, such that it supports editing.

For each of the attributes in the IFC Entities in the Concept, the values to be assigned are identified in the business rules. Are the attributes Required or Optional? Is the attribute referencing a Select or Enumerated type? In the latter case, what values are allowed? If the attribute is a Description or Name, are there conventions to be followed? These types of issues are to be captured in the Business Rules (also called Implementation Agreements).

An example of a Part P-21 instance file with the Concept IFC entities is presented, showing an example of how the attribute values are to be populated. It should include all the Entities defined in this Concept and the links between them. For a reasonable overview of the P-21 file format for reading/writing IFC files, see: http://en.wikipedia.org/wiki/ISO_10303-21.

Documents in this format are ready for depicting the leaf Concepts on the IFC Solutions Factory website. These are available for the software developers that will implement the MVD your Workgroup has specified. They are also available for other Model View Definition Workgroups that can re-use your binding specifications.

Defining the Adapter Concepts

The set of Static Concepts with IFC bindings are the building blocks used to compose Exchange Models. For use in your MVDs, they will be composed into higher level Adapter Concepts that can be re-used.

The aggregation of Static Concepts into higher level Adapter Concepts can be realized in different ways. If the target domain is a building system or material, then the Static Concepts probably provide the base units around which the Adapter Concepts are formed. (This was the case for the precast concrete MVDs, see Figure 10.) In other cases the focus may be the development of analysis or simulation capabilities (such as energy analysis or cost estimation). In these cases, the Adapter Concepts will be the analyzable units of the building with the needed properties. In the same way that a system is articulated and detailed over time, the same kind of evolution may occur in an analysis/simulation domain.

The top level Variable Concepts are primarily meant to organize and group the Adapter Concepts into a whole. These will probably be unique to the domain being used, and provide a type of indexing for the entire component Concepts, for access, and future revision and updating.

Review of Draft MVDs

While the definition of MVDs largely comprise of technical work, it is important to include two groups of stakeholders into the process. One group is the software companies who need to understand the direction and scope of the MVD implementation. Since software companies have long lead times, it is important that they have early knowledge of the effort and its expected products. Their early role is critical if the development of the desired translation software is to become part of the software company development schedules. They also have constraints based on their software and model structure that may affect details of the static binding. These issues need to be reviewed by the software companies before they are completed. The other group is the domain experts, who are seeing their specifications realized, and thus must participate with the software companies to see their needs resolved and to ensure that their MVD is implemented. Partly because of their long duration and the need for fine attention to detail, such projects have a momentum that needs to be maintained, if possible, so that the MVDs remain in the mindset of the domain experts as well as the implementers. A list of supporting documents for MVD development is given in Appendix D.

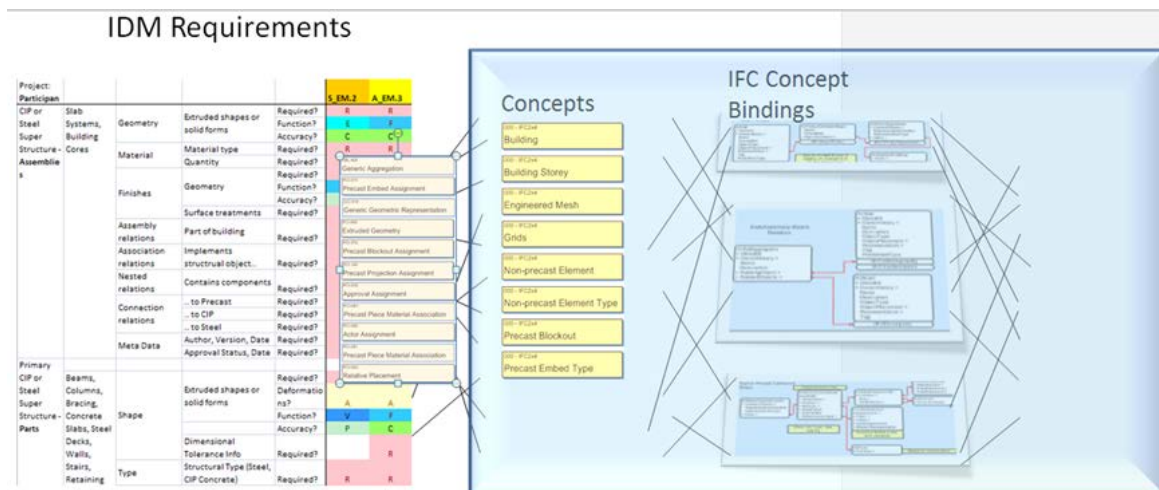


Figure 11: Relating the Developed Concepts to IDM Exchange Requirements

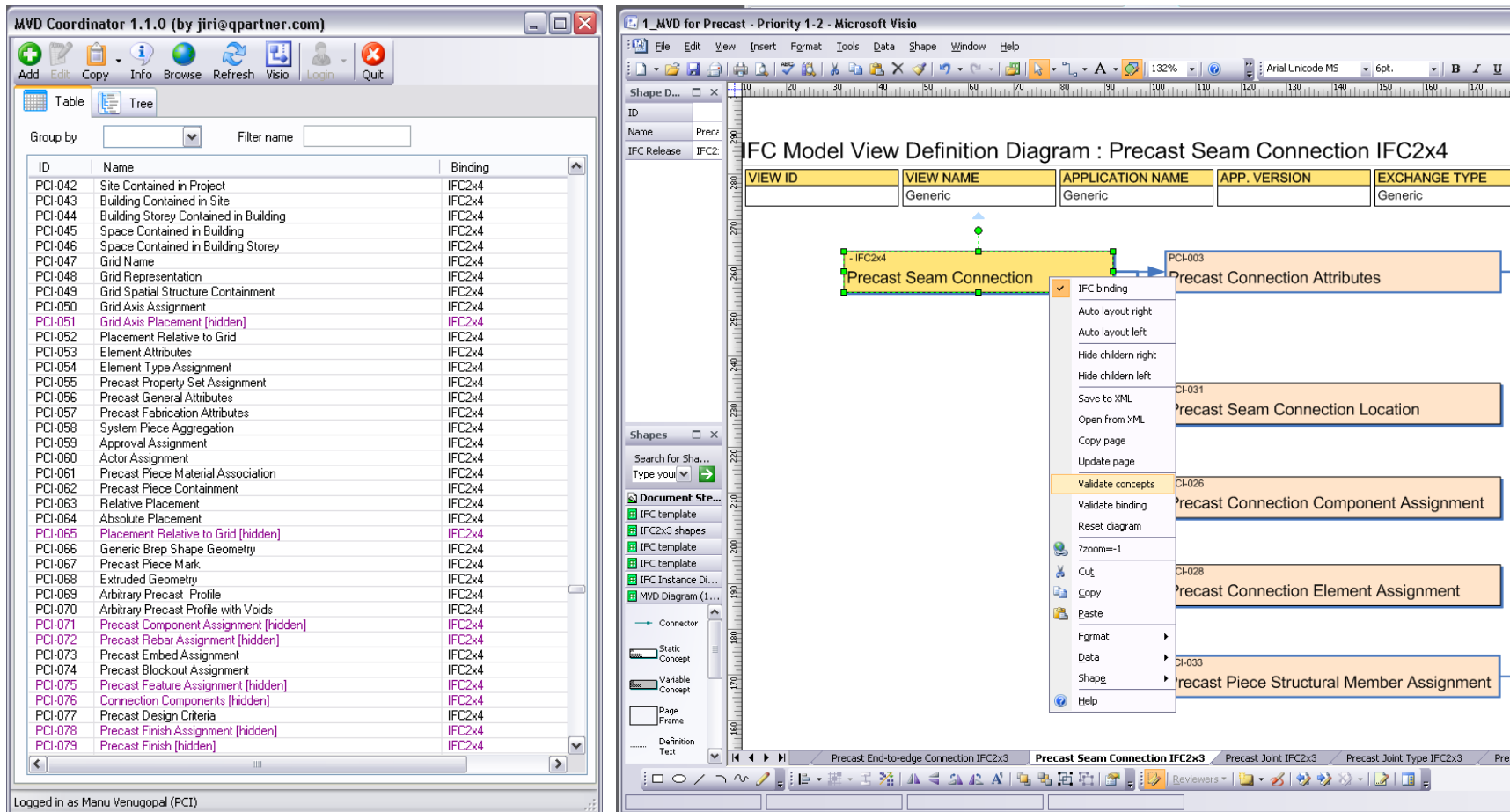
An important consideration is to define Concepts so they may be directly addressed by the Exchange Requirements of the IDM. That is, it seems possible that in the future,

the requirements generated in the IDM can be identified by selecting already defined Concepts, eliminating the current technical problem of mapping between them. This capability is diagrammed in Figure 11. This will only occur when a large set of Concepts – enough to address most mapping issues – have been defined. It also will require the definition of Concepts to be undertaken in a more rigorous and consistent manner, so they are defined by information concepts, aggregated to user-understandable needs.

2.7. Uploading the MVD and Concepts

The last step is to upload the Concepts to the IFC Library

We already explained the various components of a Model View definition. The MVD page as shown in Figure 8 is uploaded first to the IFC Solutions Factory, with the current version number and Owner information. Care should be taken in uploading the Concepts. A major benefit of Concept development and IFC Solutions Factory is the reuse of Concepts developed by one MVD group by another group. To ensure this Concepts are referenced using a unique Concept number and also the concept names. But at the same time, to maintain the integrity of the process and quality control, only the Technical Committee is given the privilege to upload a new Concept to IFC Solutions Factory. The process involves creating a placeholder for the new Concept and then a properly hyperlinked Concept document in the form of a pdf file. The placeholder mechanism allows for versioning and owner information of the Concepts, thereby ensuring that Concepts can be traced back to the owner and updated in the future if there is a need. A macro tool for Visio (refer Figure 12) was used to check that all the Concepts created and uploaded have a unique number and also to ensure that they have correct binding.



MVD Coordinator 1.1.0 (by jiri@qpartner.com)

ID	Name	Binding
PCI-042	Site Contained in Project	IFC2x4
PCI-043	Building Contained in Site	IFC2x4
PCI-044	Building Storey Contained in Building	IFC2x4
PCI-045	Space Contained in Building	IFC2x4
PCI-046	Space Contained in Building Storey	IFC2x4
PCI-047	Grid Name	IFC2x4
PCI-048	Grid Representation	IFC2x4
PCI-049	Grid Spatial Structure Containment	IFC2x4
PCI-050	Grid Axis Assignment	IFC2x4
PCI-051	Grid Axis Placement [hidden]	IFC2x4
PCI-052	Placement Relative to Grid	IFC2x4
PCI-053	Element Attributes	IFC2x4
PCI-054	Element Type Assignment	IFC2x4
PCI-055	Precast Property Set Assignment	IFC2x4
PCI-056	Precast General Attributes	IFC2x4
PCI-057	Precast Fabrication Attributes	IFC2x4
PCI-058	System Piece Aggregation	IFC2x4
PCI-059	Approval Assignment	IFC2x4
PCI-060	Actor Assignment	IFC2x4
PCI-061	Precast Piece Material Association	IFC2x4
PCI-062	Precast Piece Containment	IFC2x4
PCI-063	Relative Placement	IFC2x4
PCI-064	Absolute Placement	IFC2x4
PCI-065	Placement Relative to Grid [hidden]	IFC2x4
PCI-066	Generic Brep Shape Geometry	IFC2x4
PCI-067	Precast Piece Mark	IFC2x4
PCI-068	Extruded Geometry	IFC2x4
PCI-069	Arbitrary Precast Profile	IFC2x4
PCI-070	Arbitrary Precast Profile with Voids	IFC2x4
PCI-071	Precast Component Assignment [hidden]	IFC2x4
PCI-072	Precast Rebar Assignment [hidden]	IFC2x4
PCI-073	Precast Embed Assignment	IFC2x4
PCI-074	Precast Blockout Assignment	IFC2x4
PCI-075	Precast Feature Assignment [hidden]	IFC2x4
PCI-076	Connection Components [hidden]	IFC2x4
PCI-077	Precast Design Criteria	IFC2x4
PCI-078	Precast Finish Assignment [hidden]	IFC2x4
PCI-079	Precast Finish [hidden]	IFC2x4

1_MVD for Precast - Priority 1-2 - Microsoft Visio

IFC Model View Definition Diagram : Precast Seam Connection IFC2x4

VIEW ID	VIEW NAME	APPLICATION NAME	APP. VERSION	EXCHANGE TYPE
	Generic	Generic		Generic

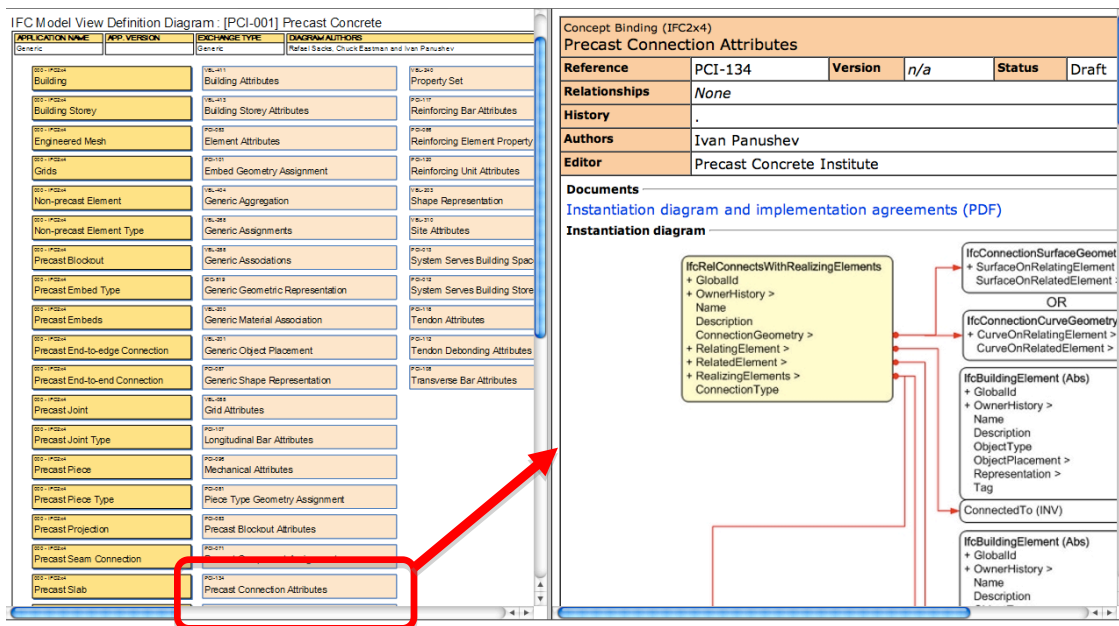
The diagram shows a central box labeled "Precast Seam Connection" connected to several other boxes representing different views or components, such as "Precast Connection Attributes", "Precast Seam Connection Location", "Precast Connection Component Assignment", "Precast Connection Element Assignment", and "Precast Piece Structural Member Assignment". A context menu is open over the "Precast Seam Connection" box, showing options like "IFC binding", "Auto layout right", "Auto layout left", "Hide children right", "Hide children left", "Save to XML", "Open from XML", "Copy page", "Update page", "Validate concepts", "Validate binding", "Reset diagram", "7zoom=-1", "Cut", "Copy", "Paste", "Format", "Data", "Shape", and "Help".

Figure 12: MVD Coordinator tool to create and check consistency of Concepts listed in an MVD.

It is important to ensure that the MVD file uploaded and the Concept document refers to the same unique Concept numbers. Otherwise, the link will be broken. The MVD coordinator tool shown in Figure 13 is used to ensure that all the Concepts are correctly numbered and linked. The Concepts present in the MVD under review are copied onto the clipboard from the Coordinator tool and then by right clicking on the MVD page in Visio, we get an option to ‘Validate the Concept’. This helps the reviewer to get a listing of Concepts that are missing or referenced incorrectly.

If all the listed Concepts are validated in this manner then we are ready to upload to the server. If the MVD and the Concept are linked correctly in this manner then by clicking on the Concept in MVD page will open the binding document side-by-side on the IFC Solutions Factory webpage. For example, these two entities; the Precast MVD and the Concept document for Precast Connection Attribute are shown in Figure 13. The Concept is given an identifying number of PCI-134 and the MVD is embedded with the following link –

<http://www.blis-project.org/IAI-MVD/reporting/showConcept.php?CREF=PCI-134>.



The image shows a screenshot of the IFC Model View Definition Diagram (MVD) interface. On the left, a table lists various concepts under the application name '[PCI-001] Precast Concrete'. The table has columns for Application Name, IFC Version, Exchange Type, and Diagram/Authors. A red box highlights the entry for 'Precast Connection Attributes' (PCI-134). A red arrow points from this entry to the right-hand side of the image, which displays the 'Concept Binding (IFC2x4) Precast Connection Attributes' page. This page includes a table with fields for Reference (PCI-134), Version (n/a), Status (Draft), Relationships (None), History, Authors (Ivan Panushev), and Editor (Precast Concrete Institute). Below this is a section for 'Documents' with a link to 'Instantiation diagram and implementation agreements (PDF)'. The main area shows an 'Instantiation diagram' with a central box for 'IfcRelConnectsWithRealizingElements' and several other boxes representing related IFC entities like 'IfcConnectionSurfaceGeometry' and 'IfcBuildingElement (Abs)'. A red box highlights the 'Precast Connection Attributes' entry in the table on the left, and a red arrow points from it to the 'Instantiation diagram' on the right.

Figure 13: Example of how MVD page is hyperlinked to Concept Document.

3. PHASE 3: CONSTRUCT

Phase 3 Overview: The purpose behind developing model views for a particular domain is that they will be implemented by software developers in a robust way and then utilized by project teams. Without effective implementation, all the previous work will remain academic. The effectiveness of the IDM and then the MVD is determined according to whether they define an effective set of interfaces for building model and related information exchange. Thus it is critical to bring into the process the relevant software development companies so they are engaged in the MVD definition process.

This Construct phase addresses this engagement with software companies and the development of the information they need to implement the specified software. We have already gotten far down the road, by determining what information should be included in an exchange, aggregating a market segment with a vested interest in a set of exchanges. Here is where we close the loop and make it happen.

3.1. Facilitate Software Product Implementation

Like most product efforts, software companies plan and generate their products according to a schedule. While every software company is different, they generally work from high-level market specifications of target results to detailed implementation specifications and outline documentation, then to implementation, then quality control testing. Since the NBIMS effort will be working alongside this process, it is beneficial to work to integrate within each company's schedules and procedures, to the degree practical.

At this stage, we have already prepared effective documentation in a format with which the software developers are familiar. It is detailed and precise (at least supposed to be). However, they need test cases of building models (or building model parts) that they can build for testing exporting, to see if they can export the data needed in an exchange. They also need example IFC instance data files that represent building

models (or building model parts) that conform to the MVD specifications of the new capabilities, to test whether they can import the data correctly into their systems.

Website support of test models needs to be provided. The website needs to allow downloading of IFC P-21 files (the project instance file format for IFC) allowing software companies to use the test file for reading purposes. Also needed are simple graphics representations, such as DXF or 2D drawings, and associated text, of project segments that incorporate model parts that require one or more Concepts for their representation in an IFC P-21 file. These project segments are to be modeled within the particular vendor's software application, then exported, to see if the system will export an IFC model with the required structure for the targeted Concepts. A set of test files and documentation of model segments should be developed to cover every Concept, and for any variations within Concepts. How will these Concept tests be undertaken? Let's back up a little to articulate what we are trying to do.

3.2. Unit Testing and General Testing

Real world model instances of some projects come in widely varying forms. They can be made anywhere within the prescribed lifecycle. They can be complete structures or only small components. They have differing levels of detail. There is no practical way to exhaustively test for all the possible conditions, at the project level. However, smaller scale testing of individual units can be exhaustive. This is generally the strategy being developed by the buildingSMART community (and widely used in electronic testing). The intention (and requirement) is to unit test every Concept that has been developed within an NBIMS effort. Later on, many of these concepts will have previously been tested by different software companies, so that these need not be re-tested again, only the new ones².

² Testing of new versions of a software package supporting IFC involves both testing of previously supported exchanges and also any new functionality supported by the release. Here we only focus on the

A current effort by the buildingSMART organization is the development of rigorous methods for testing and certification of translators, especially those that are Model Views. Different but similar test sites are being developed. The first was developed by the Institute for Advanced Building Informatics (IABI), Germany, led by Rasso Steinmann, <http://87.106.252.103/apex/f?p=101:1:2778425439471030>. This service is currently focused on testing for the Coordination View, as defined by buildingSMART international. IABI also anticipates future testing of MVDs. The second testing service, also called a BIM Validation Service, was developed by Digital Alchemy, led by Richard See, at <http://digitalalchemypro.com/html/services/IfcBimValidationService.html>. This service is focused on MVD Concept based testing. This means that a suite of unit tests are run for each Concept in the MVD, on every object instance in the file being tested. Once a user is registered, they simply select the MVD against which their building model should be validated (tested) and upload the BIM file. Detailed test results are returned to the user via email.

Both tools are accessed through application server sites avia the Web. Both are expected to improve test results reporting over time. Both sites have stated their intent to provide BIM validation for MVDs as defined in NBIMS.

Development of Test Files

First, we summarize the type of testing regime or structure that should be used in all cases. It is based on the assumption that testing sites are available. Then we explore to some degree the nature of those tests.

Validation testing of model exchanges can be broken into four levels:

testing of new functionality and assume that the software company's internal testing will deal with the previously tested and validated import and export functionality.

- a) Checking the syntax and structure of project exchange files for conformance to the IFC standard (IFC 2x3, or 2x4 etc.) this validation only applies to the export functionality of any given BIM software tool. It is not useful to test import routines this way, as import does not generate data that can be externally tested.
- b) Checking the objects in a project exchange file, as well as their properties and relationships for conformance to the bindings stipulated for them in the relevant MVD document. This test validates that the tested application can generate an exchange file with the required objects, and that these satisfy the rules of the bindings in terms of relations and attributes. The bindings for a set of Concepts are aggregated into different ways for different MVD exchanges. Thus conformance testing is performed separately for each exchange. This too is an export functionality test.
- c) Checking the import functionality of a BIM software tool for its ability to properly import the full set of concepts defined in an MVD. This can be done using a predetermined set of IFC test files that aggregate sample instances of all the Concept sets defined in the MVD. Since each possible exchange exploits a certain subset of Concepts, any given BIM software tool export function can be tested for a given exchange by testing its import of a subset of the IFC test files. This test applies to unit testing.
- d) Checking the completeness of the contents of a project exchange file (objects, parameters, and their values) between two applications, to ensure that the exchange contains all of the information required for the given exchange by the definitions of the Information Delivery Manual (IDM). This check can only be performed within the context of a precast construction project, as it check content within project context. It is an export and import test.

Concept Level Testing

The first requirement is to test each Concept within the MVDs being specified. This means that a P-21 file that embodies one or more Concepts is available. These Unit tests are not likely to be real project examples, but rather contrived tests that vary instances to address the combinations allowed by the exchange model. An example is

shown in Figure 14. It must include the required IFC Entities and satisfy at least one set of business rules to be used for import testing. To facilitate initial implementation and testing, the Technical team developed fifteen test case models. These required careful definition of the IFC files, often including manual coding of multiple lines of Part-21 files, to represent the new target output being specified. The test models need to be developed with reference to the implementation priorities received from the Advisory Committee and their software counterparts.



```

/*Precast Joint Type Profiling Geometry*/
#1247= IFCEXTRUDEDAREASOLID(#1239,#1244,#33,3657.6);
#1250= IFCSTYLEITEM(#1247, (#1234), 'Name');
#1254= IFCSHAPEPRESENTATION(#40, 'Body', 'SweptSolid', (#1247));
#1260= IFCPRODUCTDEFINITIONSHAPE('', '', (#1254));

/*Precast Joint Attributes*/
#1264= IFCFASTENER('1BAW8f00QPQ34cDZ4mD3am', #20, 'JOINT', 'D1', 'D1', #1227, #1260, 'TS_27669868');
#1283= IFCFASTENERTYPE('1jSk3Iq6H5gQkqk0c$ f$YR', #20, 'D1', $, $, $, $, $);

/*Joint Quantities*/
#1306= IFCQUANTITYLENGTH('Length', $, $, 3657.6);
#1308= IFCQUANTITYAREA('OuterSurfaceArea', $, $, 0.28950865);
#1310= IFCQUANTITYVOLUME('NetVolume', $, $, 0.0017698029);
#1312= IFCQUANTITYWEIGHT('NetWeight', $, $, 0.12388699);
#1314= IFCELEMENTQUANTITY('2ML73oG1L2u00CpCtwZeN1', #20, 'BaseQuantities', $, $,
(#1306, #1308, #1310, #1312));

/*Material Association to Fastener*/
#1319= IFCMATERIAL('MISCELLANEOUS/Insulation');
#1322= IFCRELACCREGATES('3XmgeqQP27eb1UbQpudOd', #20, $, $, #46, (#56));
#1324= IFCRELACCREGATES('1jYB390t14n9Wy_dFgh64k', #20, $, $, #56, (#69));
#1326= IFCRELACCREGATES('1ZpUmnz2DC4eQXFoLshfNH', #20, $, $, #69, (#82));
#1328= IFCRELCONTAINEDINSPATIALSTRUCTURE('3$cxzBugv1wBjRjD2Ab0m8', #20, $, $, (#1264, #657, #158), #82);
#1330=
IFCRELDEFINESBYPROPERTIES('0xI_OvJ5LEERTCoUc0j0m7', #20, 'NameRelDefByProperties', 'DescriptionRelDef
ByProperties', (#158), #199);
#1332=
IFCRELDEFINESBYPROPERTIES('1jS2Em0fv9uAdC1sUpkMy', #20, 'NameRelDefByProperties', 'DescriptionRelDef
ByProperties', (#657, #158), #216);
#1334=
IFCRELDEFINESBYPROPERTIES('ORmvmri0j8k9iqdixQYwTh', #20, 'NameRelDefByProperties', 'DescriptionRelDef
ByProperties', (#657), #684);

/*Precast Joint Attributes*/
#1338=
IFCRELDEFINESBYPROPERTIES('2zQL883czEu0EFUlpIRYsx', #20, 'NameRelDefByProperties', 'DescriptionRelDef
ByProperties', (#1264), #1314);
#1340= IFCRELDEFINESBYTYPE('lnJA7pusL0w9MwPpy7qTZe', #20, $, $, (#657, #158), #177);
#1342= IFCRELDEFINESBYTYPE('3doUjdg5r8WaoEmjSc5vaD', #20, $, $, (#1264), #1283);
#1344= IFCRELASSOCIATESMATERIAL('29Ww99e11EWwpgq9CT4Ndm', #20, $, $, (#158), #228);
#1346= IFCRELASSOCIATESMATERIAL('OtcCt4vNct1Vgh0Fze$ctBYA', #20, $, $, (#657), #699);

/*Material Association to Fastener*/
#1348= IFCRELASSOCIATESMATERIAL('OduQwoHMT98gZA_iocIQYr', #20, $, $, (#1264), #1319);
#1350= IFCPRESENTATIONLAYERASSIGNMENT('TS_1 Phase 1', $, $, (#1254, #637, #130), $);

#1360= IFCCARTESIANPOINT({6000.,0.,0.});
#1370= IFCCARTESIANPOINT({6000.,0.,3300.});
#1380= IFCPOLYLINE({#1360, #1370});
#1390= IFCCONNECTIONCURVEGEOMETRY(#1380, $);
#1400= IFCRELCONNECTSWITHREALIZINGELEMENTS('2t99E1vvjCSfo0A4jQczRu', #20, 'J-1', 'Logical
Joint', #1390, #158, #657, (#1264), 'Precast Joint');

```

Figure 14: The P-21 file segment dealing with precast joints.

Figure 15 shows the set of test files and the list of Concepts tested in each file in a tabular form. The columns outline the functionality covered in a particular test case; the rows identify the Concepts that the test file includes and will address for testing. The

important observation is that the fifteen test cases cover all the Concepts that have been defined and provide an initial base for software implementation.

Using the methods described above, test files are being developed and debugged according to the following process:

- Create initial test file from a BIM application by exporting relevant elements to IFC 2x3
- Modify manually the Part21 to reflect the Concept requirements from Figure 14 and the IFC 2x4 binding documents developed by the technical committee
- Iterate between making changes and verifying the files integrity with any Part-21 file checker, after changes for Concept
- Validate the resulting file against the IFC coordination view schema, correcting any syntax and structural errors discovered
- Verify the test file is importable without error into any IFC file viewer
- Modify the IFC 2x4 binding documents if necessary based on results and analysis from the test file development process

	A	B	C	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	MVDs Tested			Piece	Piece, Type	Piece, Slab	Piece, Projection, Blockout	Piece, Embed, Type	Joint, Joint Type, Opening and filler	End-to-end, End-to-edge, Seam	Rebar, Aggregation, Cage	Tendon	Standard Mesh, Topping/Wash	Engineered Mesh, Topping/Wash	Sandwich Wall Panel	Finish Patch, Surface Treatment
2	Features Tested			Extruded TT, Absolute Placement	Hollowcore Absolute Placement, Material	Slab, Hollowcore	Column, Projection, Blockout, Library	Column, Embed, Type	Two Wall Panels, opening, filler, joint	Two TTs, one beam, one column, three connections	Column, Rebar, Aggregation, Cage	Extruded TT, Tendon	Slab, Standard Mesh, Topping, Slab aggregation	Hollowcore Slab, Engineered Mesh, Topping, Slab aggregation	Sandwich Wall, concrete wythe	One wall panel, Finish Patch, Surface Treatment
3			Test Files	File 3	File 4	File 5	File 6	File 7	File 8	File 9	File 10	File 11	File 12	File 13	File 14	File 15
4	Concept	Number	# of Tests													
5	Precast Slab Aggregation	PCI-040	3			1							1	1		
6	Site Contained in Project	PCI-042	15	1	1	1	1	1	1	1	1	1	1	1	1	1
7	Building Contained in Site	PCI-043	15	1	1	1	1	1	1	1	1	1	1	1	1	1
8	Building Storey Contained in Building	PCI-044	15	1	1	1	1	1	1	1	1	1	1	1	1	1
9	Space Contained in Building	PCI-045	3								1					
10	Space Contained in Building Storey	PCI-046	3								1					
11	Grid Name	PCI-047	3								1					
12	Grid Representation	PCI-048	3								1					
13	Grid Spatial Structure Containment	PCI-049	3								1					
14	Grid Axis Assignment	PCI-050	3								1					
15	Placement Relative to Grid	PCI-052	2								1					
16	Element Attributes	PCI-053	14	1	1	1	1	1	1	1	1	1	1	1	1	1
17	Element Type Assignment	PCI-054	10	1	1	1			1	1		1	1	1	1	1
18	Precast Property Set Assignment	PCI-055	4	1							1	1				
19	Precast General Attributes	PCI-056	4	1							1	1				
20	Precast Fabrication Attributes	PCI-057	5	1							1	1			1	
21	System Piece Aggregation	PCI-058	4			1							1	1	1	
22	Approval Assignment	PCI-059	12	1	1	1		1	1	1	1		1	1	1	1
23	Actor Assignment	PCI-060	12	1	1	1		1	1	1	1		1	1	1	1
24	Precast Piece Material Association	PCI-061	8		1	1			1	1			1	1	1	1
25	Precast Piece Containment	PCI-062	5				1	1					1	1	1	

Figure 15: A coverage table identifying for each test file (columns) and the Concepts that the file incorporates (rows).

The testing for export is quite different. In order to export an MVD that satisfies the EM requirements, the test determines whether a user can define a model that, when exported, will satisfy the requirements defined for the MVD. A model description that is to be built within the authoring tool needs to be defined, in a manner that represents the intentions of the design. An example is shown in Figure 16. The test models should be as simple as possible, in the sense of not carrying additional design information that will need to be filtered out in the testing. A second level of testing, which is much stronger, is whether an application can, first read a file into its native structure that satisfied the requirements of the exchange model, and then export that model in IFC, in a form that matches the requirements that were embedded in the import file. This is worth considering only after the import test has been passed.

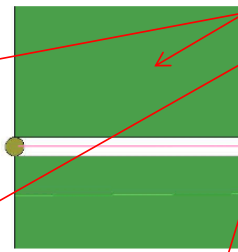
The business cases that Concept(s) must satisfy also need to be carefully documented. The business rules consist of:

- The correct types of object entities, for shapes, finishes, properties, processes, assemblies, and these are structured according to the specification
- Relations that are structured according to the schema, with proper entities connected by the relations
- Uses the specified ENUMERATED and SELECT types
- The correct values of all the attributes as specified in the MVD

An example is shown in Appendix B, at the end of this report. As the test cases are defined, they must cover all the Concepts that need testing and all the different business rules that may be applied. We recorded these in a table, as shown in Figure 16. Every concept must be tested at least once, and the various possible alternatives business rules need also to be covered; many are tested multiple times. The complete part-21 test file used to generate Figures 15 and 16 are provided for reference in Appendix C.



Implementation Package Drawings, Instructions, IFC definitions



- Precast Joint According to Concepts:**
- Precast Joint Attributes
 - Precast Joint Element Assignment
 - Precast Joint Location
 - Precast Joint Type Assignment
 - Precast Joint Type Attributes
 - Precast Joint Type Profiling Geometry

MVDs Tested	Test Files	File #	Joint, joint type, Opening and filler
			Two Wall Panels, opening, filler, joint
Site Contained in Project	PCI-012	1	1
Building Contained in Site	PCI-082	1	1
Building Storey Contained in Building	PCI-093	1	1
Element Attributes	PCI-059	1	1
Element Type Assignment	PCI-074	1	1
Approval Assignment	PCI-079	1	1
Actor Assignment	PCI-080	1	1
Precast Piece Material Association	PCI-061	1	1
Extruded Geometry	PCI-086	1	1
Precast Piece Type Attributes	PCI-088	1	1
Piece Type Geometry Assignment	PCI-081	1	1
Openings: Blockout Representation	PCI-087	1	1
Precast Joint Attributes	PCI-127	1	1
Precast Joint Element Assignment	PCI-118	1	1
Precast Joint Location	PCI-129	1	1
Precast Joint Type Assignment	PCI-150	1	1
Precast Joint Type Attributes	PCI-123	1	1
Precast Joint Type Profiling Geometry	PCI-132	1	1
Concepts Tested per File			10
Author:			Ivan

PCI Input Test File 8

Two Precast Wall Panels
Openings
Joint

Figure 16: The specification for a wall panel design incorporating a set of Concepts.

The procedures for unit testing are stringent and detailed, requiring careful composition of test files. The export file can be checked: does it include the expected entities within the Concept? Do the Entity fields carry the correct relation information? These conditions are checked on the two validation websites by automated routines, set up for the different Concepts. The technical team is responsible for defining these tests. The tests that are not satisfied are reported for correction by the software implementers. The tests are iterated until all the checks pass.

The import test that reads the P-21 test file is harder to validate. Are the intended objects represented correctly in the importing application as native objects? Did it correctly import the attributes? Most importantly, did the translator read in the GUID and other management data? The only way currently to make these checks is human inspection.

It is often proposed that an effective test for import is to export a model into IFC, then import it again, and check whether the project data for the two files are logically the same. The difficulty of the test is that such round trips in practice are rare. An architectural design application cannot carry all the information that a fabrication application carries, and vice versa, because they have quite different internal representations for the same information. In most cases, then, the Read and Write MVDs have different target Concepts.

Aggregated testing:

Most test cases developed for unit testing incorporate multiple Concepts. Thus they also identify if there are interaction effects between the rules associated for one Concept and other ones in the same test file. For example, it could happen that two different Concepts have a rule that require a “Description” field be filled in, but in different ways. The test file with the two Concepts will make this conflict apparent (and lead to revision of the Concepts).

It is important to review all the Concepts carefully to verify there are no interaction effects. For each pair of Concepts that may interact in any way, a test case should be developed. It is best to do this at the time that test cases are being programmed. Adjusting the set of test cases so they address different business conditions in combination can allow the initially defined test files to also address all, or at least most, pairwise Concept interactions.

After successful pairwise testing of Concept interactions, the next level of testing is pairwise exchanges. As testing proceeds, software companies are encouraged to exchange files and test them. This is the last stage before field testing.

Implementer's Conference Calls:

A procedural aspect of carrying out the testing is to hold bi-weekly or other scheduled conference calls with the implementer software companies. These calls are to address problems as they arise, ambiguities of testing, review of the test files and their structure, and other issues that arise during testing. The calls are usually led by the technical advisory team.

3.3. SW Certification Testing and Reporting

An important component of validation and certification is a public testing and reporting site. This is or should be a component of any public testing site. It allows any organization to upload a P-21 test file and to validate the file against any registered MVD's requirements. Discrepancies are reported and errors are recorded on the website. Thus the status of any software's ability to export a model view is publicly documented. Software companies can announce what Model Views they believe their software supports. The public testing provides an open and public means to check the status and open issues with regard to any model exchange within any software package. This public reporting and review of translation errors is sometimes called

"self-certification", in that it is carried out by the software companies, with public oversight.

3.4. Submission of a Draft MVD for Review

It is expected that multiple MVDs will be generated in parallel activities. Most parallel activities will be recognized through their publication on the IFC Solutions Factory website, which will make the activities visible. However, an important role of the buildingSMART organization is to provide coordination across these efforts, especially when there are “ownership” questions. All standards are public and not owned by anyone but are in the public domain. The buildingSMART organization is responsible for smoothing out any conflicts arising from overlapping model view scopes.

Also, the buildingSMART organization is responsible for seeing that the procedures it has set forward are followed, to support the logic and transparency of the MVD results. These will become more regular as the NBIMS matures.

It must be noted that the buildingSMART reviews should be pursued throughout the NBIMS process. Finding conflicts at this stage is like finding spatial conflicts during layout at the construction site – it’s at the last minute and the most expensive time.

PHASE 4: DEPLOY

The last stage of the National BIM Standard is deployment by the software companies and field testing by users. The support for deployment is ongoing and support transfers to the responsibility of the software companies.

4.1. Product Specific BIM Guides

Each BIM authoring tool has its own method for defining objects, assigning relationships (connection, embedding) and for assigning attributes. At this level, the way data is defined in any given BIM tool may be quite different from the way it is done in other BIM tools, even to produce the same output data. Thus it is necessary to provide a **BIM-tool Specific User Guide** for defining particular constructs for translation. This is an important level of documentation needed to support BIM standard exchanges.

Each software vendor is responsible for their user guides, which are ultimately an extension of the company's application user guide. The user guides themselves are increasingly on-line and accessible directly from within the application.

An outline for the functionality addressed in the user guide should be produced in Phase 2. It is this functionality that each vendor needs to show how to support. Examples are such details as: assigning properties to the bolts in a connection, to define whether a beam is represented as manufactured or as placed in a building with pre- or post-tensioning and deflections; how to assign various specialized properties. Only with this level of documentation will users be able to utilize the exchanges.

4.2. BIM Exchange and Data Validation

The domain team should consider documenting early projects that use the new model views. Part of this is to identify problems and possible errors. As the field tests proceed, publicity about these projects should be undertaken, in general support of the

efforts of participants and the supporting industry group. This is part of the public announcement about the capabilities realized through the NBIMS effort.

5. Post Script

Workflows evolve. What was a manually undertaken task, such as copying work orders, or doing a space comparison for clash checking, are automated and the workflow changes. This process of change will accelerate as the potential for automation is applied to all aspects of the design and building process. Thus exchange requirements will evolve and the process of updating them will be a new (but easier) undertaking. After this major effort, it is hard to accept that workflow analysis is not a one-time effort, but rather an incremental one. "The path has been plowed and the trip has been initiated. It is unclear where the path ends".

GLOSSARY

exchange model: An exchange model is made up of a use case embedded in a process model and a set of exchange requirements for the use case. It lays out the user specifications for a use case.

data object: Data is organized into groups, called data objects. All the information in a data object refers to the same thing. Thus a process object may have a name, start time, duration, resource requirements, which are all attributes of the data object.

schema: the structures used to carry data electronically. In a paper world, the equivalent of a schema is a form, with content boxes. Attributes such as ‘first name’, or ‘nationality’ distinguishes what the text in a box means. There are also relations between the data, such as the relation between the name of a respondent and the names of next-of-kin, with separate addresses for each. Possibly the relation is asked for- son, granddaughter, etc. A product model schema is the specific structure by which a computer can read and interpret the data it is carrying.

use case: One or more exchanges that define the communication between two AEC roles that addresses a particular issue

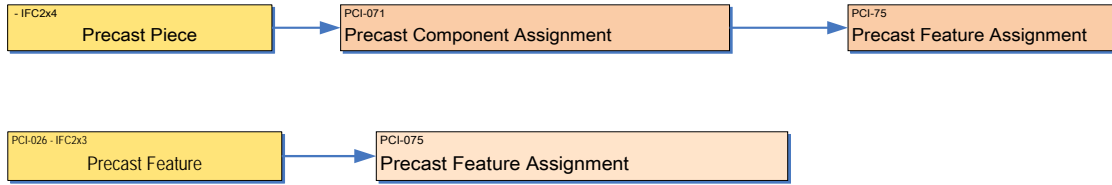
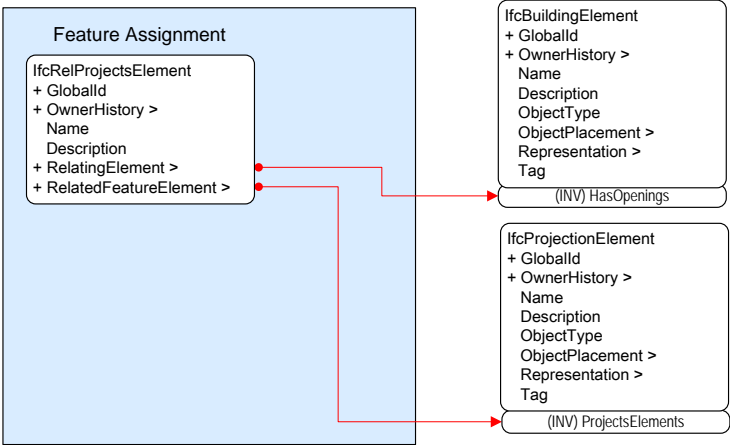
workgroup: A team of people that in this case are brought together to address a domain of workflow exchanges.

workflow: the path undertaken in some piece of work. Here, workflows are the communication and coordination required to accomplish some tasks that also involves BIM data exchange.

6. BIBLIOGRAPHY

- Aram, V., Eastman, C.M., Sacks, R., Panushev, I., and Venugopal, M., (2010). 'Introducing a New Methodology to Develop the Information Delivery Manual for AEC Projects', Applications of IT in the AEC Industry, CIB W78 Workshop Proceedings, Cairo, Egypt, November 2010, Virginia Tech, 10 pps.
- Hietanen, J., (v1: 2005/v2: 2008). 'IFC Model View Definition Format', buildingSMART International Technical Management (ITM) meeting #38, Amsterdam, NV, February 2008, buildingSMART, 29 pps.
- See, R., Karshoej, J., (2011). 'Integration of IDM and MVD', buildingSMART International Technical Management (ITM) meeting #47, Abu Dhabi, UAE, March 2011, buildingSMART, 53 pps.

APPENDIX A: Sample IFC Binding document

IFC Release Specific Concept Description (IFC 2x4)					
Precast Feature Assignment					
Reference	<Reference field>	Version	0.7	Status	Draft
Relationships	Provides a Boolean addition to a precast piece, such as a corbel. It is a part of the connection through IfcRelConnectsWithRealizingElements, as well as a feature of the main building element.				
History					
Authors	Rafael Sacks				
Document Owner	Precast/Prestressed Concrete Institute (cvsacks@technion.ac.il)				
Usage in view definition diagram					
 <pre> graph LR A["- IFC2x4 Precast Piece"] --> B["PCI-071 Precast Component Assignment"] B --> C["PCI-75 Precast Feature Assignment"] D["PCI-026 - IFC2x3 Precast Feature"] --> C </pre>					
Instantiation diagram					
 <pre> classDiagram class FeatureAssignment { +IfcRelProjectsElement +GlobalId +OwnerHistory > Name Description +RelatingElement > +RelatedFeatureElement > } class IfcBuildingElement { +GlobalId +OwnerHistory > Name Description ObjectType ObjectPlacement > Representation > Tag } class IfcProjectionElement { +GlobalId +OwnerHistory > Name Description ObjectType ObjectPlacement > Representation > Tag } FeatureAssignment --> IfcBuildingElement : (INV) HasOpenings FeatureAssignment --> IfcProjectionElement : (INV) ProjectsElements </pre>					
Implementation agreements					
IfcRelProjectsElement					
Attribute	Implementation agreements				
GlobalId	Must be provided				
OwnerHistory	Must be provided, but may contain dummy data				
Name	<Open>				



Description	<Open>
RelatingElement	The hosting precast piece (subtype of precast IfcBuildingElement).
RelatedFeatureElement	The feature element (IfcProjectionElement) (corbels, shelves, etc.)

Example: Part21 file

Precast Beam, 40x80 cm

```
#108= IFCCARTESIANPOINT((0.,0.));
#112= IFCDIRECTION((1.,0.));
#116= IFCAXIS2PLACEMENT2D(#108,#112);
#119= IFCRECTANGLEPROFILEDEF(.AREA.,'400*800',#116,400.,800.);
#120= IFCDIRECTION((0.,-1.,0.));
#124= IFCDIRECTION((-1.,0.,0.));
#128= IFCCARTESIANPOINT((6000.,0.,0.));
#132= IFCAXIS2PLACEMENT3D(#128,#124,#120);
#135= IFCEXTRUDEDAREASOLID(#119,#132,#33,5780.);
#138= IFCSTYLEDITEM(#135,(#106),'Name');
#142= IFCSHAPE REPRESENTATION(#40,'Body','SweptSolid',(#135));
#148= IFCPRODUCTDEFINITIONSHAPE("",( #142));
#152= IFCBEAM('1AH9bc00001p4oD3OmC3at',#20,'BEAM','400*800','400*800',#99,#148,'TS_1667');
```

Precast Blockout – to accommodate corbel

```
#214= IFCAXIS2PLACEMENT3D(#92,#33,#25);
#217= IFCLocalPLACEMENT(#79,#214);
#220= IFCAXIS2PLACEMENT2D(#108,#112);
#223= IFCRECTANGLEPROFILEDEF(.AREA.,",#220,300.,420.);
#224= IFCDIRECTION((0.,0.,-1.));
#228= IFCCARTESIANPOINT((520.,0.,-250.));
#232= IFCAXIS2PLACEMENT3D(#228,#124,#224);
#235= IFCEXTRUDEDAREASOLID(#223,#232,#33,320.);
#238= IFCSHAPE REPRESENTATION(#40,'Body','SweptSolid',(#235));
#244= IFCPRODUCTDEFINITIONSHAPE("",( #238));
#248= IFCOPENINGELEMENT('1AH9bc00008Z4oD3OmC3at',#20,"",'Recess',#217,#244,"");
#269= IFCRELVOIDSELEMENT('14s_88t61DDAWL8lgoQ$vk',#20,"",#152,#248);
```

Precast Blockout – 6x8cm to accommodate shear pin

```
#270= IFCAXIS2PLACEMENT3D(#92,#33,#25);
#273= IFCLocalPLACEMENT(#79,#270);
#276= IFCAXIS2PLACEMENT2D(#108,#112);
#279= IFCRECTANGLEPROFILEDEF(.AREA.,",#276,60.,80.);
#280= IFCCARTESIANPOINT((330.,0.,-105.));
#284= IFCAXIS2PLACEMENT3D(#280,#33,#120);
#287= IFCEXTRUDEDAREASOLID(#279,#284,#33,510.);
#290= IFCSHAPE REPRESENTATION(#40,'Body','SweptSolid',(#287));
#296= IFCPRODUCTDEFINITIONSHAPE("",( #290));
#300= IFCOPENINGELEMENT('1AH9bc0000AZ4oD3OmC3at',#20,"",#273,#296,"");
#321= IFCRELVOIDSELEMENT('0aj1PHxYbCRg0czFQ$vo6H',#20,"",#152,#300);
```

Precast Column 40x40 cm



```
#338= IFCAXIS2PLACEMENT2D(#108,#112);
#341= IFCRECTANGLEPROFILEDEF(.AREA.,'400*400',#338,400.,400.);
#342= IFCARTESIANPOINT((0.,0.,4000.));
#346= IFCAXIS2PLACEMENT3D(#342,#224,#124);
#349= IFCEXTRUDEDAREASOLID(#341,#346,#33,4000.);
#352= IFCSTYLEITEM(#349,('336'),'Name');
#356= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#349));
#362= IFCPRODUCTDEFINITIONSHAPE(" ",(#356));
#366= IFCCOLUMN('1AH9bc00000p4oD3OmC3at',#20,'COLUMN','400*400','400*400',#329,#362,'TS_1635');
#385= IFCCOLUMNTYPE('2JpEbDhDD3URR4xIAGV2A9',#20,'400*400',$,$,$,$,$,$,.NOTDEFINED.);
```

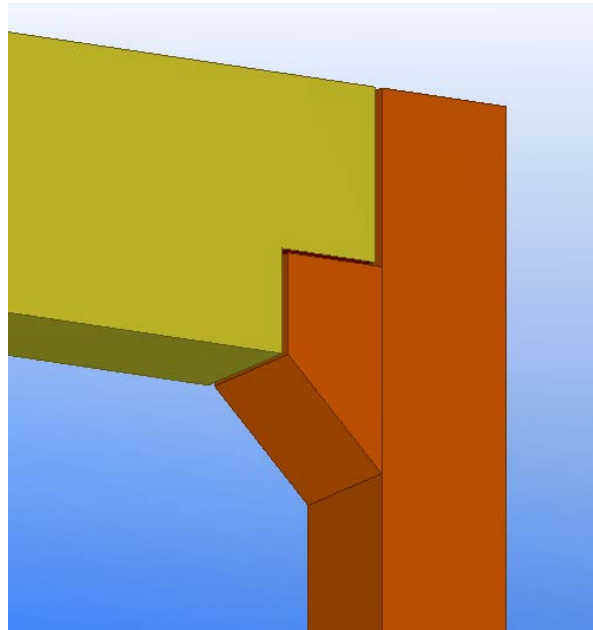
Precast Feature – Corbel

```
#508= IFCARTESIANPOINT((0.,590.));
#512= IFCARTESIANPOINT((300.,290.));
#516= IFCARTESIANPOINT((300.,0.));
#520= IFCARTESIANPOINT((0.,0.));
#524= IFCPOLYLINE((#508,#512,#516,#520,#508));
#528= IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,'PLT400*300',#524);
#529= IFCARTESIANPOINT((0.,0.,-200.));
#533= IFCAXIS2PLACEMENT3D(#529,#33,#25);
#536= IFCEXTRUDEDAREASOLID(#528,#533,#33,400.);

#543= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#536));
#549= IFCPRODUCTDEFINITIONSHAPE(" ",(#543));
#553= IFCPROJECTIONELEMENT('1AH9bc00005p4oD3OmC3at',#20,'Corbel','Corbel cast with column','Precast Feature',#99,#549,'TS_2331',.CORBEL.);
#555= IFCRELPROJECTSELEMENT('2D89NMFVzDoeR6bcf1DEQ9',#20,'Feature 1', 'Precast Feature Relationship',#366,#553);
```

Connection Geometry and Relationship

```
#699= IFCARTESIANPOINT((0.,210.,4000.));
#701= IFCAXIS2PLACEMENT3D(#699,#25,#29);
#703= IFCPLANE(#701);
#799= IFCARTESIANPOINT((0.,200.,4000.));
#801= IFCAXIS2PLACEMENT3D(#799,#33,#25);
#803= IFCPLANE(#701);
#901= IFCCONNECTIONSURFACEGEOMETRY(#703,#803);
#903= IFCRELCONNECTSWITHREALIZINGELEMENTS('2t99ElvjjCSfoOA4jQczRu',#20,$,$,#901,#152,#366,(#553),'Corbel');
```



Precast Feature Example - Corbel on Column

This document uses the official IFC Model View Definition Format version 1.1.0. of the IAI (www.iai-international.org)

The content of this document has to be certified by the IAI before becoming part of an official IFC Model View Definition.

APPENDIX B: Precast Joint Model Exchange Validation

The required concepts and rules for validating a precast joint in a model exchange are defined here. The corresponding concepts as defined in PCI Model View Definitions are listed here. A list of the minimum subset of IFC entities required to satisfy the precast joint and the corresponding relationships are also provided. The business rules to be satisfied as part of validating the ifc entities for precast specific needs are provided as 8 major conditions. Each of these has a set of sub-rules to be satisfied. It is assumed that each of the entities and relationships listed here are completed without any missing references or pointers (Usually performed as part of any IFC syntax checking tool).

A. Concept mapping from IfcSolutionsFactory

1. Precast Joint Attributes	PCI-147
2. Precast Joint Element Assignment	PCI-148
3. Precast Joint Location	PCI-149
4. Precast Joint Type Assignment	PCI-150
5. Precast Joint Type Attributes	PCI-151
6. Precast Joint Type Profiling Geometry	PCI-152

B. List of IFC entities required for the model exchange

Entities:

IfcFastener,
IfcFastenerType,
Valid subtypes of IfcBuildingElement,
IfcMaterial,

IfcElementQuantity,
IfcConnectionCurveGeometry,
IfcLine(IfcPolyline or IfcCompositeCurve),
IfcRepresentationMap (depending on the IfcFastenerType. See rules),

Relationships:

IfcRelDefinesByType,
IfcRelConnectsWithRealizingElements,
IfcRelAssociatesMaterial
IfcRelDefinesbyProperties

C. Business Rules

Condition 1: At least one instance of IfcFastener, which satisfies the following attribute values.

Should have 8 attributes

First 2 attributes should compulsorily point to GUID and Owner History

Third and fourth attributes are optional

Fifth attribute should be ObjectType and contain string 'Precast Joint' and there should be a reference linked to IfcFastenerType using IfcRelDefinesByType

Also, in IfcRelConnectsWithRealizingElements the Connectingtype attribute should match the ObjectType. i.e 'Precast Joint'

Sixth attribute is object placement and should point to a valid placement concept

Seventh attribute is representation and should point to a valid geometry concept

Eighth is 'tag' – optional

Condition 2: Should attach an IfcMaterial to the IfcFastener (joint) through IfcRelAssociatesMaterial, for example, bituminous rubber for compression seal.

Condition 3: Area, Volume, and Weight should be attached to the IfcFastener using IfcElementQuantity through IfcRelDefinesbyProperties

Additional property sets (Optional) if existing should be attached through IfcRelDefinesbyProperties

Condition 4: IfcFastener should be assigned to subtypes of IfcBuildingElement through IfcRelConnectsWithRealizingElements. The following checks should be satisfied

There should be at least one instance of IfcRelConnectsWithRealizingElements such that it connects IfcFastener to 2 different building elements.

The RelatingElement attribute should point to a physical piece which is a precast element and should not be a piece type

The RelatedElement attribute should point to any physical piece, precast or non-precast or members of other structural system

The connection geometry should define the line of joint and point to IfcConnectionCurveGeometry

The connecting type should be 'precast joint'

Condition 5: there should be at least one instance of IfcConnectionCurveGeometry

The curve on relating element attribute should be used to limit the location and extent of the joint and should point to a valid IfcLine or IfcPolyline or IfcCompositeCurve

Curve on related element should be null? – the curve is identical on both elements

Condition 6: Check for the existence of IfcFastenerType. If present it should satisfy the following rules.

The IfcFastenerType should be linked to IfcFastener through IfcRelDefinesByType

GUID, OwnerHistory – mandatory

Elementtype – should be provided.

Possible values include:

- Vertical open-drained (Slotted neoprene baffle plus vertical air-seal)
- Horizontal open-drained (profiled with flashing and horizontal air-seal)
- Face Sealed
- Compression Seal with gasket
- Compression Seal with flexible material
- Other

The applicable occurrence can have three values namely, based on which the representation maps attribute also changes.

Refer following table.

Value of applicable occurrence	Corresponding value of representation maps
None	Null –neither element is profiled
RelatingOnly	Point to a single representation map (first element is profiled)
RelatedOnly	Point to a single representation map (second element is profiled)
Both	Point to two representation maps

Condition 7: If FastenerType is present, then there should be corresponding instance of IfcRelDefinesByType relationship

The relatedobjects should point to instances of IfcFasteners only

The relating type should point to instances of IfcFastenerType only

Condition 8: Based on the value of Applicable Occurrences in FastenerType there should be 0,1, or 2 instances of valid IfcRepresentationMap

Should point to a valid IfcRepresentationMap (as an initial test case). In reality this should map to a geometric set of 2D curves that define the cross-section profiling of the pieces on either side of the joint.

D. Sample Part 21 File

A sample Part-21 test file is given in Appendix C



APPENDIX C: Sample Part-21Test File

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('ViewDefinition [CoordinationView,
QuantityTakeOffAddOnView]'),'2;1');
FILE_NAME('C:\\TeklaStructuresModels\\Precast NBIMS Test Model 3\\PCI IFC
test file 8.ifc','2009-12-17T13:27:48','(TD-CV/rafael)','Structural
Designer'),'EXPRESS Data Manager version:20070116','Tekla Structures 16.0
Build:563554/27.10.2009, IFC Export Version:119/Oct 23 2009','');
FILE_SCHEMA(('IFC2X3'));
ENDSEC;

DATA;
#1= IFCPERSON('TD-CV/rafael','Undefined',,$,$,$,$,$,$);
#3= IFCORGANIZATION($,'Tekla Corporation',,$,$,$);
#7= IFCPERSONANDORGANIZATION(#1,#3,$);
#8= IFCAPPLICATION(#3,'16.0','Tekla Structures','Multi material modeling');
#9= IFCSIUNIT(*,.LENGTHUNIT.,.MILLI.,.METRE.);
#10= IFCSIUNIT(*,.AREAUNIT.,$.SQUARE_METRE.);
#11= IFCSIUNIT(*,.VOLUMEUNIT.,$.CUBIC_METRE.);
#12= IFCSIUNIT(*,.MASSUNIT.,.KILO.,.GRAM.);
#13= IFCSIUNIT(*,.TIMEUNIT.,$.SECOND.);
#14= IFCSIUNIT(*,.PLANEANGLEUNIT.,$.RADIAN.);
#15= IFCSIUNIT(*,.SOLIDANGLEUNIT.,$.STERADIAN.);
#16= IFCSIUNIT(*,.THERMODYNAMICTEMPERATUREUNIT.,$.DEGREE_CELSIUS.);
#17= IFCSIUNIT(*,.LUMINOUSINTENSITYUNIT.,$.LUMEN.);
#18= IFCUNITASSIGNMENT((#9,#10,#11,#12,#13,#14,#15,#16,#17));
#20= IFCOWNERHISTORY(#7,#8,$,.ADDED.,$,,$,$,1261049268);
#21= IFCCARTESIANPOINT((0.,0.,0.));
#25= IFCDIRECTION((1.,0.,0.));
#29= IFCDIRECTION((0.,1.,0.));
#33= IFCDIRECTION((0.,0.,1.));
#37= IFCAXIS2PLACEMENT3D(#21,#33,#25);
#40= IFCGEOMETRICREPRESENTATIONCONTEXT('Body','Model',3,1.0000000E-5,#37,$);
#43= IFCGEOMETRICREPRESENTATIONCONTEXT('BoundingBox','Model',3,1.0000000E-
5,#37,$);
#46= IFCPROJECT('3Sj3c8AR52ZwdXihqTohdw',#20,'WINCHESTER MEDICAL CTR EAST
P/S','Description','Object type','LongName','Phase',(#40,#43),#18);
#53= IFCLOCALPLACEMENT($,#37);
#56=
IFCSITE('2lNlIU_hP19f5MUyULqtU1',#20,'Undefined',,$,$,#53,$,$,.ELEMENT.,$,0.
,$,$);
#66= IFCLOCALPLACEMENT(#53,#37);
#69=
IFCBUILDING('2_eJKBDIf10Om09XwUSOFJ',#20,'Undefined',,$,$,#66,$,$,.ELEMENT.,$,
,$,$);
#79= IFCLOCALPLACEMENT(#66,#37);

#82=
IFCBUILDINGSTOREY('1BVUDwk9T7AhzEMacRQyB1',#20,'Undefined',,$,$,#79,$,$,.ELEMEN
T.,$);
#92= IFCCARTESIANPOINT((152.4,-1828.8,8432.8));
#96= IFCAXIS2PLACEMENT3D(#92,#33,#29);
#99= IFCLOCALPLACEMENT(#79,#96);
#102= IFCCOLOURRGB('Light Green',0.3019608,0.89803922,0.3019608);
```



```
#103=
IFCSURFACESTYLERENDERING(#102,0.,$,,$,$,$,IFCNORMALISEDRAIOMEASURE(0.00390625
),IFCSPECULAREXONENT(10.),.NOTDEFINED.);
#104= IFCSURFACESTYLE('',.POSITIVE.,(#103));
#106= IFCPRESENTATIONSTYLEASSIGNMENT((#104));
#108= IFCCARTESIANPOINT((1828.8,0.));
#112= IFCDIRECTION((1.,0.));
#116= IFCAXIS2PLACEMENT2D(#108,#112);
#119= IFCRECTANGLEPROFILEDEF(.AREA.,'12"X144"',#116,3657.6,304.8);
#120= IFCAXIS2PLACEMENT3D(#21,#33,#25);
#123= IFCEXTRUDEDAREASOLID(#119,#120,#33,6337.3);
#126= IFCSTYLEDEITEM(#123,(#106),'Name');
#130= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#123));
#136= IFCCARTESIANPOINT((0.,0.));
#140= IFCCARTESIANPOINT((3657.6,0.));
#144= IFCPOLYLINE((#136,#140));
#148= IFCSHAPEREPRESENTATION(#40,'Axis','Curve2D',(#144));
#154= IFCPRODUCTDEFINITIONSHAPE('','',(#130,#148));
#158=
IFCWALLSTANDARDCASE('18sLIo000AeJ4oCZ8oDZWm',#20,'140012','12"X144"', '12"X144
','',#99,#154,'Piece_Mark_Test');

#177=
IFCWALLTYPE('21As$GxTv9P96JaIOcStup',#20,'12"X144"',$,,$,$,$,$,$,.NOTDEFINED.)
;

#183= IFCPROPERTY SINGLEVALUE('Class','Class',IFCIDENTIFIER('3'),$);
#187= IFCPROPERTY SINGLEVALUE('Finish','Finish',IFCIDENTIFIER(''),$);
#191=
IFCPROPERTY SINGLEVALUE('AssemblyId','AssemblyId',IFCIDENTIFIER('SW8(?)'),$);
#195=
IFCPROPERTY SINGLEVALUE('Part_Position','Part_Position',IFCIDENTIFIER('Concret
e109(?)'),$);
#199=
IFCPROPERTYSET('0Ba8gJct92zOaL7cMWNn3y',#20,'Pset_Tekla_General','Pset_Tekla_
General',(#183,#187,#191,#195));
#204= IFCQUANTITYLENGTH('Width',$, $,304.8);
#206= IFCQUANTITYAREA('GrossFootprintArea',$, $,0.90580464);
#208= IFCQUANTITYLENGTH('Length',$, $,6337.3);
#210= IFCQUANTITYVOLUME('NetVolume',$, $,6.7957094);
#212= IFCQUANTITYLENGTH('Height',$, $,6337.3);
#214= IFCQUANTITYWEIGHT('NetWeight',$, $,16328.513);
#216=
IFCELEMENTQUANTITY('2W6G0$ia133e5W67uULoe_',#20,'BaseQuantities',$, $,($204,#2
06,#208,#210,#212,#214));
#221= IFCMATERIAL('CONCRETE/5000');
#224= IFCMATERIALLAYER(#221,304.8,$);
#226= IFCMATERIALLAYERSET((#224),'Wall: Insitu CONCRETE/5000 304.800000');
#228= IFCMATERIALLAYERSETUSAGE(#226,.AXIS2.,.POSITIVE.,-152.4);
#229= IFCAXIS2PLACEMENT3D(#92,#33,#29);
#232= IFCLOCALPLACEMENT(#79,#229);
#235= IFCAXIS2PLACEMENT2D(#136,#112);
#238= IFCRECTANGLEPROFILEDEF(.AREA.,'',#235,685.8,1130.3);
#239= IFCDIRECTION((-1.,0.,0.));
#243= IFCDIRECTION((0.,0.9999446,-0.0105257));
#247= IFCCARTESIANPOINT((1828.8,-183.21248,5938.3479));
#251= IFCAXIS2PLACEMENT3D(#247,#243,#239);
#254= IFCEXTRUDEDAREASOLID(#238,#251,#33,355.61688);
#257= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#254));
#263= IFCPRODUCTDEFINITIONSHAPE('','',(#257));
```



```
#267=
IFCOPENINGELEMENT('19OqxW0017Rp4oCp4oDp8s',#20,'','','Recess',#232,#263,'');
#288= IFCRELVOIDSELEMENT('1Ytr4py25E$gCf50pmDdYc',#20,'',' ',#158,#267);
#289= IFCAXIS2PLACEMENT3D(#92,#33,#29);
#292= IFCLOCALPLACEMENT(#79,#289);
#295= IFCAXIS2PLACEMENT2D(#136,#112);
#298= IFCRECTANGLEPROFILEDEF(.AREA.,',',#295,152.4,304.8);
#299= IFCDIRECTION((0.,0.0105257,0.9999446));
#303= IFCDIRECTION((0.,-0.9999446,0.0105257));
#307= IFCCARTESIANPOINT((2603.5,177.79858,3248.5292));
#311= IFCAXIS2PLACEMENT3D(#307,#303,#299);
#314= IFCEXTRUDEDAREASOLID(#298,#311,#33,355.61688);
#317= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#314));
#323= IFCPRODUCTDEFINITIONSHAPE('','',(#317));
#327=
IFCOPENINGELEMENT('19OqxW001AP34oCp4oDpSm',#20,'','','Recess',#292,#323,'');
#348= IFCRELVOIDSELEMENT('3KkRwzUO93DRZ385K5OR_o',#20,'',' ',#158,#327);
#349= IFCAXIS2PLACEMENT3D(#92,#33,#29);
#352= IFCLOCALPLACEMENT(#79,#349);
#355= IFCAXIS2PLACEMENT2D(#136,#112);
#358= IFCRECTANGLEPROFILEDEF(.AREA.,',',#355,152.4,304.8);
#359= IFCCARTESIANPOINT((1054.1,177.79859,3248.5292));
#363= IFCAXIS2PLACEMENT3D(#359,#303,#299);
#366= IFCEXTRUDEDAREASOLID(#358,#363,#33,355.61688);
#369= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#366));
#375= IFCPRODUCTDEFINITIONSHAPE('','',(#369));
#379=
IFCOPENINGELEMENT('19OqxW001APp4oCp4oDpSm',#20,'','','Recess',#352,#375,'');
#400= IFCRELVOIDSELEMENT('33Bx$6Erv6LP1KpdulhaYh',#20,'',' ',#158,#379);
#401= IFCAXIS2PLACEMENT3D(#92,#33,#29);
#404= IFCLOCALPLACEMENT(#79,#401);
#407= IFCAXIS2PLACEMENT2D(#136,#112);
#410= IFCCIRCLEPROFILEDEF(.AREA.,',',#407,101.6);
#411= IFCDIRECTION((0.,-1.,0.));
#415= IFCCARTESIANPOINT((1054.1,177.8,5524.501));
#419= IFCAXIS2PLACEMENT3D(#415,#411,#25);
#422= IFCEXTRUDEDAREASOLID(#410,#419,#33,355.6);
#425= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#422));
#431= IFCPRODUCTDEFINITIONSHAPE('','',(#425));
#435=
IFCOPENINGELEMENT('19ahHb0007_J4oCpGpDJCn',#20,'','','Recess',#404,#431,'');
#456= IFCRELVOIDSELEMENT('0Eytq4za90gQ5wGKPjvFIQ',#20,'',' ',#158,#435);
#457= IFCAXIS2PLACEMENT3D(#92,#33,#29);
#460= IFCLOCALPLACEMENT(#79,#457);
#463= IFCAXIS2PLACEMENT2D(#136,#112);
#466= IFCCIRCLEPROFILEDEF(.AREA.,',',#463,101.6);
#467= IFCCARTESIANPOINT((2603.5,177.8,5524.501));
#471= IFCAXIS2PLACEMENT3D(#467,#411,#25);
#474= IFCEXTRUDEDAREASOLID(#466,#471,#33,355.6);
#477= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#474));
#483= IFCPRODUCTDEFINITIONSHAPE('','',(#477));
#487=
IFCOPENINGELEMENT('19ahHb0007$34oCpGpDJCn',#20,'','','Recess',#460,#483,'');
#508= IFCRELVOIDSELEMENT('3armAliFf5AvKns7DAk91o',#20,'',' ',#158,#487);
#509= IFCAXIS2PLACEMENT3D(#92,#33,#29);
#512= IFCLOCALPLACEMENT(#79,#509);
#515= IFCAXIS2PLACEMENT2D(#136,#112);
#518= IFCCIRCLEPROFILEDEF(.AREA.,',',#515,101.6);
#519= IFCCARTESIANPOINT((1054.1,177.8,2273.301));
#523= IFCAXIS2PLACEMENT3D(#519,#411,#25);
```



```
#526= IFCEXTRUDEDAREASOLID(#518,#523,#33,355.6);
#529= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#526));
#535= IFCPRODUCTDEFINITIONSHAPE('','',(529));
#539=
IFCOPENINGELEMENT('19ahHb0007$P4oCpGpDJCo',#20,'','','Recess',#512,#535,'');
#560= IFCRELVOIDSELEMENT('2onnZrcCn8196ig2MnVzbc',#20,'','','#158,#539);
#561= IFCAXIS2PLACEMENT3D(#92,#33,#29);
#564= IFCLOCALPLACEMENT(#79,#561);
#567= IFCAXIS2PLACEMENT2D(#136,#112);
#570= IFCCIRCLEPROFILEDEF(.AREA.,'',#567,101.6);
#571= IFCCARTESIANPOINT((2603.5,177.8,2273.301));
#575= IFCAXIS2PLACEMENT3D(#571,#411,#25);
#578= IFCEXTRUDEDAREASOLID(#570,#575,#33,355.6);
#581= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#578));
#587= IFCPRODUCTDEFINITIONSHAPE('','',(581));
#591=
IFCOPENINGELEMENT('19ahHb00080Z4oCpGpDJCo',#20,'','','Recess',#564,#587,'');
#612= IFCRELVOIDSELEMENT('0_CmABIS95eu30jIFEvAxx',#20,'','','#158,#591);
#613= IFCCARTESIANPOINT((152.4,-1828.8,965.2));
#617= IFCAXIS2PLACEMENT3D(#613,#33,#29);
#620= IFCLOCALPLACEMENT(#79,#617);
#623= IFCAXIS2PLACEMENT2D(#108,#112);
#626= IFCRECTANGLEPROFILEDEF(.AREA.,'12"X144"',#623,3657.6,304.8);
#627= IFCAXIS2PLACEMENT3D(#21,#33,#25);
#630= IFCEXTRUDEDAREASOLID(#626,#627,#33,7442.2);
#633= IFCSTYLEITEM(#630,(#106),'Name');
#637= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#630));
#643= IFCPOLYLINE((#136,#140));
#647= IFCSHAPEREPRESENTATION(#40,'Axis','Curve2D',(#643));
#653= IFCPRODUCTDEFINITIONSHAPE('','',(637,#647));
#657=
IFCWALLSTANDARDCASE('18sLIo000AdJ4oCz8oDZWm',#20,'140012','12"X144"',#12"X144"',#620,#653,'Piece_Mark_Test');
#676=
IFCPROPERTYSINGLEVALUE('AssemblyId','AssemblyId',IFCIDENTIFIER('SW6(?)'),$);
#680=
IFCPROPERTYSINGLEVALUE('Part_Position','Part_Position',IFCIDENTIFIER('Concret
e107(?)'),$);
#684=
IFCPROPERTYSET('0F7Dsf9Hn0gQQE98z7151C',#20,'Pset_Tekla_General','Pset_Tekla_
General',(#183,#187,#676,#680));
#689= IFCQUANTITYAREA('GrossFootprintArea',$$,$,0.9290304);
#691= IFCQUANTITYLENGTH('Length',$$,$,7442.2);
#693= IFCQUANTITYVOLUME('NetVolume',$$,$,8.1915233);
#695= IFCQUANTITYLENGTH('Height',$$,$,7442.2);
#697= IFCQUANTITYWEIGHT('NetWeight',$$,$,19682.33);
#699= IFCMATERIALLAYERSETUSAGE(#226,.AXIS2..POSITIVE..-152.4);
#700= IFCAXIS2PLACEMENT3D(#613,#33,#29);
#703= IFCLOCALPLACEMENT(#79,#700);
#706= IFCAXIS2PLACEMENT2D(#136,#112);
#709= IFCRECTANGLEPROFILEDEF(.AREA.,'',#706,152.4,304.8);
#710= IFCCARTESIANPOINT((2603.5,177.79858,4213.7292));
#714= IFCAXIS2PLACEMENT3D(#710,#303,#299);
#717= IFCEXTRUDEDAREASOLID(#709,#714,#33,355.61688);
#720= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#717));
#726= IFCPRODUCTDEFINITIONSHAPE('','',(720));
#730=
IFCOPENINGELEMENT('19OqxW001AM34oCp4oDpSm',#20,'','','Recess',#703,#726,'');
#751= IFCRELVOIDSELEMENT('3pKQrXnazFXOi3x5DBgF63',#20,'','','#657,#730);
#752= IFCAXIS2PLACEMENT3D(#613,#33,#29);
```




```
#755= IFCLOCALPLACEMENT(#79,#752);
#758= IFCAXIS2PLACEMENT2D(#136,#112);
#761= IFCRECTANGLEPROFILEDEF(.AREA.,',',#758,152.4,304.8);
#762= IFCCARTESIANPOINT((1054.1,177.79859,4213.7292));
#766= IFCAXIS2PLACEMENT3D(#762,#303,#299);
#769= IFCEXTRUDEDAREASOLID(#761,#766,#33,355.61688);
#772= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#769));
#778= IFCPRODUCTDEFINITIONSHAPE('','',(#772));
#782=
IFCOPENINGELEMENT('19OqxW001Amp4oCp4oDpSm',#20,'','','Recess',#755,#778,'');
#803= IFCRELVOIDSELEMENT('0QzPggp$WfDoPDVP58jIlfX',#20,'','','#657,#782);
#804= IFCAXIS2PLACEMENT3D(#613,#33,#29);
#807= IFCLOCALPLACEMENT(#79,#804);
#810= IFCAXIS2PLACEMENT2D(#136,#112);
#813= IFCRECTANGLEPROFILEDEF(.AREA.,',',#810,152.4,304.8);
#814= IFCCARTESIANPOINT((2603.5,177.79859,7464.9292));
#818= IFCAXIS2PLACEMENT3D(#814,#303,#299);
#821= IFCEXTRUDEDAREASOLID(#813,#818,#33,355.61688);
#824= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#821));
#830= IFCPRODUCTDEFINITIONSHAPE('','',(#824));
#834=
IFCOPENINGELEMENT('19OqxW001ANZ4oCp4oDpSm',#20,'','','Recess',#807,#830,'');
#855= IFCRELVOIDSELEMENT('0FNyNLn8b0fPt0R9JimWCN',#20,'','','#657,#834);
#856= IFCAXIS2PLACEMENT3D(#613,#33,#29);
#859= IFCLOCALPLACEMENT(#79,#856);
#862= IFCAXIS2PLACEMENT2D(#136,#112);
#865= IFCRECTANGLEPROFILEDEF(.AREA.,',',#862,152.4,304.8);
#866= IFCCARTESIANPOINT((1054.1,177.79859,7464.9292));
#870= IFCAXIS2PLACEMENT3D(#866,#303,#299);
#873= IFCEXTRUDEDAREASOLID(#865,#870,#33,355.61688);
#876= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#873));
#882= IFCPRODUCTDEFINITIONSHAPE('','',(#876));
#886=
IFCOPENINGELEMENT('19OqxW001AOJ4oCp4oDpSm',#20,'','','Recess',#859,#882,'');
#907= IFCRELVOIDSELEMENT('1OZtw7H1b818GBE_qDPfjq',#20,'','','#657,#886);
#908= IFCAXIS2PLACEMENT3D(#613,#33,#29);
#911= IFCLOCALPLACEMENT(#79,#908);
#914= IFCAXIS2PLACEMENT2D(#136,#112);
#917= IFCRECTANGLEPROFILEDEF(.AREA.,',',#914,152.4,304.8);
#918= IFCCARTESIANPOINT((2603.5,177.79858,784.72915));
#922= IFCAXIS2PLACEMENT3D(#918,#303,#299);
#925= IFCEXTRUDEDAREASOLID(#917,#922,#33,355.61688);
#928= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#925));
#934= IFCPRODUCTDEFINITIONSHAPE('','',(#928));
#938=
IFCOPENINGELEMENT('19T7xa0006JZ4oCp8pDpat',#20,'','','Recess',#911,#934,'');
#959= IFCRELVOIDSELEMENT('2o1Dd$JNf5Lg0bCG8BwaRm',#20,'','','#657,#938);
#960= IFCAXIS2PLACEMENT3D(#613,#33,#29);
#963= IFCLOCALPLACEMENT(#79,#960);
#966= IFCAXIS2PLACEMENT2D(#136,#112);
#969= IFCRECTANGLEPROFILEDEF(.AREA.,',',#966,152.4,304.8);
#970= IFCCARTESIANPOINT((1054.1,177.79858,784.72915));
#974= IFCAXIS2PLACEMENT3D(#970,#303,#299);
#977= IFCEXTRUDEDAREASOLID(#969,#974,#33,355.61688);
#980= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#977));
#986= IFCPRODUCTDEFINITIONSHAPE('','',(#980));
#990=
IFCOPENINGELEMENT('19T7xa0006KJ4oCp8pDpat',#20,'','','Recess',#963,#986,'');
#1011= IFCRELVOIDSELEMENT('1VyJbtGEN0UfHijsQtECy8',#20,'','','#657,#990);
#1012= IFCAXIS2PLACEMENT3D(#613,#33,#29);
```



```

#1015= IFCLOCALPLACEMENT(#79,#1012);
#1018= IFCAXIS2PLACEMENT2D(#136,#112);
#1021= IFCCIRCLEPROFILEDEF(.AREA.,',',#1018,101.6);
#1022= IFCCARTESIANPOINT((1054.1,177.8,6489.701));
#1026= IFCAXIS2PLACEMENT3D(#1022,#411,#25);
#1029= IFCEXTRUDEDAREASOLID(#1021,#1026,#33,355.6);
#1032= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#1029));
#1038= IFCPRODUCTDEFINITIONSHAPE('','',(#1032));
#1042=
IFCOPENINGELEMENT('19ahHb00081J4oCpGpDJCo',#20,'','','Recess',#1015,#1038,'')
;
#1063= IFCRELVOIDSELEMENT('3mQaHf$MjAgPulmrOO5_2F',#20,'','','#657,#1042);
#1064= IFCAXIS2PLACEMENT3D(#613,#33,#29);
#1067= IFCLOCALPLACEMENT(#79,#1064);
#1070= IFCAXIS2PLACEMENT2D(#136,#112);
#1073= IFCCIRCLEPROFILEDEF(.AREA.,',',#1070,101.6);
#1074= IFCCARTESIANPOINT((2603.5,177.8,6489.701));
#1078= IFCAXIS2PLACEMENT3D(#1074,#411,#25);
#1081= IFCEXTRUDEDAREASOLID(#1073,#1078,#33,355.6);
#1084= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#1081));
#1090= IFCPRODUCTDEFINITIONSHAPE('','',(#1084));
#1094=
IFCOPENINGELEMENT('19ahHb0008234oCpGpDJCo',#20,'','','Recess',#1067,#1090,'')
;
#1115= IFCRELVOIDSELEMENT('0Cd00NP7P9f9qbwzi7EAOo',#20,'','','#657,#1094);
#1116= IFCAXIS2PLACEMENT3D(#613,#33,#29);
#1119= IFCLOCALPLACEMENT(#79,#1116);
#1122= IFCAXIS2PLACEMENT2D(#136,#112);
#1125= IFCCIRCLEPROFILEDEF(.AREA.,',',#1122,101.6);
#1126= IFCCARTESIANPOINT((1054.1,177.8,3238.501));
#1130= IFCAXIS2PLACEMENT3D(#1126,#411,#25);
#1133= IFCEXTRUDEDAREASOLID(#1125,#1130,#33,355.6);
#1136= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#1133));
#1142= IFCPRODUCTDEFINITIONSHAPE('','',(#1136));
#1146=
IFCOPENINGELEMENT('19ahHb00082p4oCpGpDJCo',#20,'','','Recess',#1119,#1142,'')
;
#1167= IFCRELVOIDSELEMENT('1L0nm2rDv5GBAZHrn0Lh3O',#20,'','','#657,#1146);
#1168= IFCAXIS2PLACEMENT3D(#613,#33,#29);
#1171= IFCLOCALPLACEMENT(#79,#1168);
#1174= IFCAXIS2PLACEMENT2D(#136,#112);
#1177= IFCCIRCLEPROFILEDEF(.AREA.,',',#1174,101.6);
#1178= IFCCARTESIANPOINT((2603.5,177.8,3238.501));
#1182= IFCAXIS2PLACEMENT3D(#1178,#411,#25);
#1185= IFCEXTRUDEDAREASOLID(#1177,#1182,#33,355.6);
#1188= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#1185));
#1194= IFCPRODUCTDEFINITIONSHAPE('','',(#1188));
#1198=
IFCOPENINGELEMENT('19ahHb00083Z4oCpGpDJCo',#20,'','','Recess',#1171,#1194,'')
;
#1219= IFCRELVOIDSELEMENT('0vr1RjqJv1YxBoiAONSn2W',#20,'','','#657,#1198);
#1220= IFCCARTESIANPOINT((8.6401997E-12,1828.8,8420.1));
#1224= IFCAXIS2PLACEMENT3D(#1220,#33,#411);
#1227= IFCLOCALPLACEMENT(#79,#1224);
#1230= IFCCOLOURRGB('Light Yellow',0.89803922,0.89803922,0.2);
#1231=
IFCSURFACESTYLERENDERING(#1230,0.,$,,$,$,IFCNORMALISED RATIO MEASURE(0.0039062
5),IFCSPECULAREXPONENT(10.),.NOTDEFINED.);
#1232= IFCSURFACESTYLE('',.POSITIVE.,(#1231));
#1234= IFCPRESENTATIONSTYLEASSIGNMENT((#1232));

```



```
#1236= IFCAXIS2PLACEMENT2D(#136,#112);
#1239= IFCCIRCLEPROFILEDEF(.AREA.,'D1"',#1236,12.7);
#1240= IFCCARTESIANPOINT((3657.6,0.,0.));
#1244= IFCAXIS2PLACEMENT3D(#1240,#239,#411);

/*Precast Joint Type Profiling Geometry*/
#1247= IFCEXTRUDEDAREASOLID(#1239,#1244,#33,3657.6);
#1250= IFCSTYLEDITEM(#1247,(#1234),'Name');
#1254= IFCSHAPEREPRESENTATION(#40,'Body','SweptSolid',(#1247));
#1260= IFCPRODUCTDEFINITIONSHAPE('','',(#1254));

/*Precast Joint Attributes*/
#1264=
IFCFASTENER('1BAW8f000PQ34oDZ4mD3am',#20,'JOINT','D1','',#1227,#1260,'TS_
27669868');
#1283= IFCFASTENERTYPE('1jSk3Iq6H5gQkqkOc$f$YR',#20,'D1','',,$,$,$,$,$,$);

/*Joint Quantities*/
#1306= IFCQUANTITYLENGTH('Length',,$,$,3657.6);
#1308= IFCQUANTITYAREA('OuterSurfaceArea',,$,$,0.28950865);
#1310= IFCQUANTITYVOLUME('NetVolume',,$,$,0.0017698029);
#1312= IFCQUANTITYWEIGHT('NetWeight',,$,$,0.12388699);
#1314=
IFCELEMENTQUANTITY('2ML73oG1L2u00CpCtwZeN1',#20,'BaseQuantities',,$,$,(#1306,#
1308,#1310,#1312));

/*Material Association to Fastener*/
#1319= IFCMATERIAL('MISCELLANEOUS/Insulation');
#1322= IFCRELAGGREGATES('3Xmgeq0QP27eb1UbQpuD0d',#20,$,$,#46,(#56));
#1324= IFCRELAGGREGATES('1jYB390t14n9Wy_dFgh64k',#20,$,$,#56,(#69));
#1326= IFCRELAGGREGATES('1ZpUmnz2DC4eQXFoLshfNH',#20,$,$,#69,(#82));
#1328=
IFCRELCONTAINEDINSPATIALSTRUCTURE('3$cxzBugv1wBjRjD2AbOm8',#20,$,$,(#1264,#65
7,#158),#82);
#1330=
IFCRELDEFINESBYPROPERTIES('0xI_0vJ5LEeRTCoUcOjOM7',#20,'NameRelDefByPropertie
s','DescriptionRelDefByProperties',(#158),#199);
#1332=
IFCRELDEFINESBYPROPERTIES('1jS2Em0fv9uAdClsUpkxMy',#20,'NameRelDefByPropertie
s','DescriptionRelDefByProperties',(#657,#158),#216);
#1334=
IFCRELDEFINESBYPROPERTIES('0Rmvmri0j8k9iqdixQYwTh',#20,'NameRelDefByPropertie
s','DescriptionRelDefByProperties',(#657),#684);

/*Precast Joint Attributes*/
#1338=
IFCRELDEFINESBYPROPERTIES('2zQL883czEuOEFu1pIRYsx',#20,'NameRelDefByPropertie
s','DescriptionRelDefByProperties',(#1264),#1314);
#1340=
IFCRELDEFINESBYTYPE('1nJA7pusL0w9MwPpy7qTZe',#20,$,$,(#657,#158),#177);
#1342= IFCRELDEFINESBYTYPE('3doUjdg5r8WeoEmjSc5vaD',#20,$,$,(#1264),#1283);
#1344=
IFCRELASSOCIATESMATERIAL('29WW99el1EWwpgq9CT4Ndn',#20,$,$,(#158),#228);
#1346=
IFCRELASSOCIATESMATERIAL('0tCt4vNcT1VghOFze$tBYA',#20,$,$,(#657),#699);

/*Material Association to Fastener*/
#1348=
IFCRELASSOCIATESMATERIAL('0duQwoHMT98gZA_iocIQYr',#20,$,$,(#1264),#1319);
```



```
#1350= IFCPRESENTATIONLAYERASSIGNMENT('TS_1 Phase 1',$(#1254,#637,#130),$);

#1360= IFCCARTESIANPOINT((6000.,0.,0.));
#1370= IFCCARTESIANPOINT((6000.,0.,3300.));
#1380= IFCPOLYLINE((#1360,#1370));
#1390= IFCCONNECTIONCURVEGEOMETRY(#1380,$);
#1400= IFCRELCONNECTSWITHREALIZINGELEMENTS('2t99EivvjCSfoOA4jQczRu',#20,'J-1',
'Logical Joint',#1390,#158,#657,(#1264),'Precast Joint');

ENDSEC;

END-ISO-10303-21;
```

APPENDIX D: List of supporting documents for MVD development

1. PCI BIM Project Webpage

<http://dcom.arch.gatech.edu/pcibim/>

2. Model View Definitions for Precast Concrete

http://dcom.arch.gatech.edu/pcibim/documents/Precast_MVDs_v2.1_Volume_I.pdf

3. Draft IFC Binding Documents for Precast Concrete Concepts

[http://dcom.arch.gatech.edu/pcibim/documents/Precast_MVDs_v2.1_Volume_II-
ce.pdf](http://dcom.arch.gatech.edu/pcibim/documents/Precast_MVDs_v2.1_Volume_II-
ce.pdf)

4. Information Delivery Manual for Precast Concrete

http://dcom.arch.gatech.edu/pcibim/documents/IDM_for_Precast.pdf

APPENDIX E

Possible Technical Advisor Contacts for National BIM Standards Work

Chuck Eastman

Thomas Liebich

Nick Nisbet

Ivan Panushev

Rafael Sacks

Richard See