

# A Semantic Enrichment Engine for Building Information Modeling

Michael Belsky<sup>1</sup>, Rafael Sacks<sup>1</sup>, and Ioannis Brilakis<sup>2</sup>

<sup>1</sup>Faculty of Civil and Environmental Engineering, Technion – Israel Institute of Technology, Haifa, Israel

<sup>2</sup>Department of Engineering – University of Cambridge, UK

**Abstract:** *Significant difficulties remain in exchanging information between Building Information Modeling (BIM) tools. The Industry Foundation Classes (IFC) exchange schema is too generic to capture the full semantic meaning needed for direct use by different construction project stakeholders' BIM tools. Although BIM standards that prescribe model view definitions (MVD) for domain-specific exchanges are under development, insufficient semantic definition of exchanges prevents achievement of the full potential of BIM through seamless interoperability. We propose an innovative approach for supplementing an IFC exchange file with semantically useful concepts inferred from the explicit and implicit information contained in the building model. The rule-sets used can easily be tailored for different domains. A prototype system was implemented and used to demonstrate semantic enrichment of precast concrete model files, adding joint and slab aggregation concepts. The tests illustrated the composition of rule-sets using topological, geometric and other generic operators that can be re-used for other domains.*

## 1 INTRODUCTION

The architecture, engineering and construction (AEC) industry requires a high level of collaboration between multiple different parties involved in construction projects and encompasses domains that make diverse use of building information. Ideally, the information could be easily shared, retrieved and used for various engineering simulations and analyses, such as structural stability, energy consumption, cost estimation and building code compliance checking (Eastman, Teicholz, et al. 2011). However, individual Building Information Modeling (BIM) applications store information in their native formats, imposing challenges for re-using the data in subsequent applications downstream in the workflow. The Industry Foundation Classes (IFC) product data model schema (Liebich, et al. 2006) was specifically designed to support interoperability within the AEC industry. The IFC schema is rich and extensive, but it is also complex and has redundancies because of the need to represent objects and relationships for a wide range of AEC sub-domains (Eastman, Jeong, et al. 2010). For building information model exchanges to be meaningful for importing tools, type-instance relations, aggregations, geometry and topological relationships must be defined precisely. Because there are multiple ways to represent any given semantic concept, implementers of IFC export and import subroutines in BIM software tools need clear guidance for specific information exchanges (Sacks, et al. 2010). The efforts to prepare Model View Definitions (MVDs) aim to provide this semantic clarity.

Existing BIM software tools cannot interpret geometric and topological information that is not explicitly contained in IFC models because they lack the required domain knowledge about semantics of spatial relationships and attributes. Therefore, the MVD approach assumes that software companies will develop IFC export and import subroutines tailored for each MVD. This is time consuming and costly, and difficult to justify commercially. Much of current practice therefore relies on import of models as reference geometry only, and the importer remodels the building in terms of the native objects of the importing application.

An alternative approach is to build generic expert systems that can use domain-specific rule-sets to infer the semantics of a model, thus enriching it and providing a form that can be imported and used directly. The geometric and material properties of building elements and the spatial topology relationships among them determine the way architects and engineers understand their function and expected behavior, and thus the way in which they should be modeled. The idea is to capture the ways that professionals interpret the implicit meaning in building models they view in the form of expert system rules, and then process the rule sets using an inference engine to semantically enrich building models.

The following section reviews related research and approaches to the interoperability problem, identifies open issues and positions our research in these contexts. The third section outlines the design of a semantic enrichment engine for BIM (which we call ‘SEEBIM’), including definition of the required geometry and spatial topology operators and description of a prototype implementation. Following this are illustrations of semantic enrichment inference rules designed to infer semantic constructs as defined by the US National BIM Standard (C. M. Eastman, R. Sacks, et al. 2010) for the precast concrete domain. An evaluation of the proposed approach and discussion of potential applications conclude the paper.

## 2 BACKGROUND

The development of BIM for the AEC industry has generally followed the development of product modeling in the manufacturing industry. The notions of object-oriented modeling and parametric modeling of buildings have their roots in the concepts of parametric solid modeling for generic product modeling (Sacks, R.; Eastman, C. M.; Lee, G.; 2004, Shah, J. J.; Mantyla, M.; 1995). The IFC building product model schema is based on the foundations of ISO 16739 (ISO 16739:2013 2014), including the EXPRESS language (ISO 10303-11 2014); and BIM tools embody ideas derived from parametric mechanical CAD tools. In contrast to this trend, the development of semantic enrichment to support interoperability of product models for manufacturing, as described below, has not yet been proposed for interoperability of building models.

Research efforts in semantic interpretation of building information appear to be limited to attempts to compile building models by inferring information from 2D CAD files (Babalola 2012, Noack 2001), to schematic and server-based query approaches for interrogating IFC models and to topological reasoning over IFC models (Nguyen, Oloufa and Nassar 2005, Beetz, Leeuwen and Vries 2006). Interpretation of 2D CAD drawings has proven to be a far more complex problem than initially envisaged and the demand for it is reduced once 3D models are available directly from BIM. Schematic and server-based approaches are useful, but they do not make implicit information explicit (Mazairac and Beetz 2013). The topological reasoning approaches are an important contribution toward semantic enrichment (Borrmann and Rank 2009). They are domain-specific. The efforts to-date are platform dependent and require significant programming skills as well as deep knowledge of the product model schema, and as such they have limited flexibility and require time consuming and error prone human intervention.

This chapter therefore reviews the development of interoperability for BIM and of semantic enrichment in product life-cycle management (PLM) systems.

### 2.1 Industry Foundation Classes and Model View Definitions

The Industry Foundation Classes (IFC) schema is widely recognized as the common data exchange format for interoperability within the AEC industry (Eastman, Teicholz, et al. 2011). The initial specification for IFC was compiled using the EXPRESS language, starting in 1995, to address the sharing of information to facilitate and support efficient workflows and information exchanges in the AEC industry (Leibich and Wix 1998, Liebich, et al. 2006). Managed by the buildingSMART organization, it has undergone numerous improvements and versions since then (BuildingSmart.). It is currently going through the process of becoming an International Standard, ISO 16739.

IFC entities represent geometry, relations, processes and materials, performance, fabrication, and other properties, needed for design, production and maintenance of building facility. It is a rich product model schema, but it is highly redundant and lacks formal logic rigidity. As a result, data exchanges selecting from the redundant data representations have had unacceptable problems of mismatch and reliability, and these has posed a barrier to the advance of BIM (Eastman, Jeong, et al. 2010, Olofsson, Eastman and Lee 2008). Studies on data exchanges show that without well-defined exchange model views, the exchanges are vulnerable to errors, omissions, contradictions and misrepresentation because they reduce or simplify the information (Bazjanac and Kiviniemi 2007). The results of the exchange scenarios between BIM applications have been shown to contain information losses or distortions (Palzar and Turk 2008). Most of these problems can be related to the lack of

semantic uniformity in the way BIM tools map their internal objects to and from IFC entities and properties. Performance studies of BIM data bases, to create partial models and run queries, also show a strong need for both identifying model views for specific exchanges, as well as for specifying the exchange protocols in a stricter manner (Nour 2009, Eastman, C.; Jeong, Y.; Sacks, R.; Kaner, I.; 2010). Venugopal et al. ( 2010) argue for the need for a more formal definition of IFC concepts.

These inconsistencies have led to the conclusion that domain specific model view definitions (MVD) are needed to define precisely how building model exchanges should be expressed using IFC (Final and Hietanen 2006). A layer of specificity for selecting and specifying information entities, their attributes and rules over the top of the IFC schema needs to be provided for effective exchanges. This layer is a subset of the IFC schema and when used for a particular exchange it is called a model view. A more generic definition of an MVD is 'a subset of a building product model schema that provides a critical representation of the information concepts needed for a particular information exchange in AEC workflow' (Eastman, Panushev, et al. 2011).

In 2010, buildingSMART developed the IFC Certification 2.0 procedure for the IFC 2X3 Coordination View Version 2 (CV2.0), which is an MVD intended to promote consistent and reliable implementations of the IFC specification by many software vendors across multiple software platforms. Its purpose is limited to support of clash-checking and other basic building system coordination tasks, which require exchange of geometry and part identities. These are arguably the least demanding aspects of BIM interoperability.

The National BIM Standard™ initiative (NIBS 2014) adopts the idea of facilitating information exchanges through MVDs (Hietanen and Final 2007). Its methodology defines the process for determining the appropriate information entities from the IFC schema for any particular use, representing them in a domain-specific Information Delivery Manual (IDM), and finally compiling an MVD specification. The information concepts are shared through an open website, IFC Solution Factory (See 2010).

## **2.2 Extracting partial building models**

The importance of extracting useful subsets from IFC building models, to suit domain-specific applications or data exchange requirements, has been recognized by different researchers (Beetz, van Leeuwen and de Vries 2009, Wiese, Katranuchkov and Scherer 2003). The following are some of the approaches developed for extracting information according to a certain view or deriving a subset of all entities from a building model instance:

- a graph query method, which filters out geometry and topology information (Beetz, van Leeuwen and de Vries 2009);
- a Generalized Model Subset Definition schema proposed in Wiese, Katranuchkov and Scherer (2003);
- a no-schema extraction algorithm for IFC instance models (Won, Lee and Cho 2013).

There are also server-based approaches for creating partial building models. However, the expressive power of the query languages provided by product model servers such as BiMserver (Beetz, van Berlo, et al. 2010), Jotne EDMServer (Jotne EPM Technology) or EuroStep Model Server (Eurostep) is limited: they are unable to interpret the implicit information contained in building models because they are not familiar with the spatial semantics of particular attributes and relationships. All of these approaches leave the original building model objects essentially unchanged. However, in real information exchanges between BIM tools, building entities and/or semantic constructs might need to be modified or added to a model to suit data exchange requirements. For example, containment and inverse relationships can be changed or inferred as described in Zang and Issa (2011).

## **2.3 Semantic enrichment of product models**

In the manufacturing industry, numerous product modeling technologies and product data managing tools have emerged as a result of various research efforts which aimed to capture and share product data information and knowledge. Product Data Management (PDM) systems, which were introduced in the late 1980s (Ameri and Dutta 2005), are in widespread use. More recently, efforts have been made to extend the functionality of PDM systems to provide support for the full range of activities needed in a product's lifecycle, from conception to

disposal, and have resulted in Product Lifecycle Management (PLM) systems. PDM and PLM technologies facilitate access to information and focus on product form representation, but they lack intelligence (Mocko, et al. 2004). The field of product modeling, which incorporates knowledge representation, sharing and exchanging, has been reported in many papers. Methodologies in this field can be classified in four main types: solid modeling, feature-based modeling, knowledge-based modeling and integrated modeling. Solid modeling, such as Constructive Solid Geometry (CSG) and Boundary representation (B-rep), is very common and focuses on 3D shape objects. Feature-based modeling associates functional information with form features. Knowledge-based modeling adopts artificial intelligence techniques and focuses on structuring knowledge. It has several advantages (Chen and Wei 1997) because it describes elements and relations between them, but it depends on human intervention and lacks formality. In contrast with the first three types of modeling, integrated modeling can support various applications in different product development stages. The integration is achieved using global unified modelling and mapping of models. ISO 10303 (ISO, ISO 10303 2014), the Standard for the Exchange of Product model data (STEP) is an example of such integration.

As in the AEC industry, modeling of a product over its lifecycle implies the need for interoperability of product models between software tools with a variety of domain-specific functionality and thus different internal representations. Specifically, design intent, modeled in the form of parametric constraints between objects (for example, the relationship between the diameter of an axle and the diameters of the bearings in which it is mounted) should be carried when data is transferred from one tool to another. Innovative PLM technologies have been developed that can infer such relationships from the geometry and spatial topology. Tools such as Siemens Synchronous Technology (Siemens), for example, can use data from multiple CAD systems. Given a model in B-rep the tools can process it into a feature-based CSG representation and in doing so infer parametric dimensions and constraints that can be used to position and size its objects, and control their shapes, efficiently. With the development of Semantic Web, ontology engineering has become crucial, because it provides the necessary semantics for many fields of knowledge (Noy and McGuinness 2001) and facilitates knowledge reuse and sharing. An example of using core product ontology has been reported in (Lou 2011). In this example existing semantics of product data models is integrated with core product ontology in order to build semantically enriched product model. Product models are the primary source of knowledge and information throughout different stages of product lifecycle. However, product models are usually generated in heterogeneous computing platforms and as a result the semantics are hidden in information structures of different product data formats. This imposes difficulties for seamless interoperability and efficient reuse of knowledge.

## 2.4 Summary

The complexity of modern buildings has led to the need for extensive and intensive collaboration between different construction project stakeholders that use different software tools to support their activities. As a result, reliable information exchanges between BIM tools are essential to maximize the benefits of BIM. The IFC schema, which covers a wide range of modeling information across the broad AEC industry, is the de-facto standard for model based interoperability. However, it faces significant challenges in semantic interoperability because of the intrinsic problem of the differences between the ways in which different professions conceptualize the same physical objects (e.g., consider the difference between an architectural column versus a structural column).

Existing approaches to overcoming the limitations of interoperability have so far had limited success in improving semantic interoperability. There are essentially three such approaches:

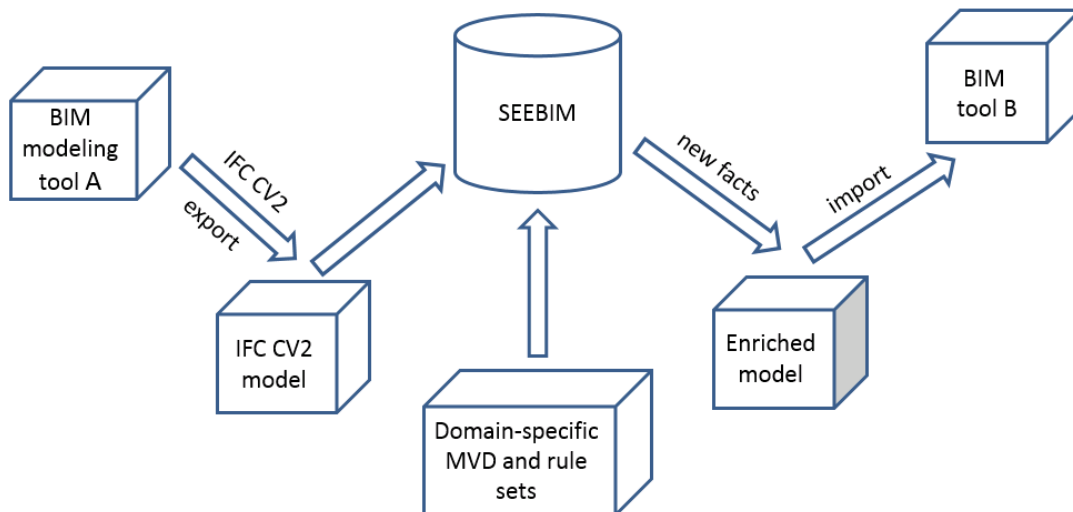
1. Rigorous compilation of domain-specific model view definitions to define granular building concepts, with the expectation that software companies will provide matching export and import subroutines. This approach appears to resurrect the original challenge of interoperability among multiple tools that gave impetus to the IFC development, which was the need for  $n/2 (n - 1)$  separate exchanges for  $n$  tools.
2. Partial model views for extracting sub-views from an IFC model.
3. Server-based querying approaches that operate on instance and schema level and are used for interrogating IFC models.

All in all, none of the existing approaches can create semantically enriched IFC model useful for receiving application. Therefore, a new solution is needed to cover the gap.

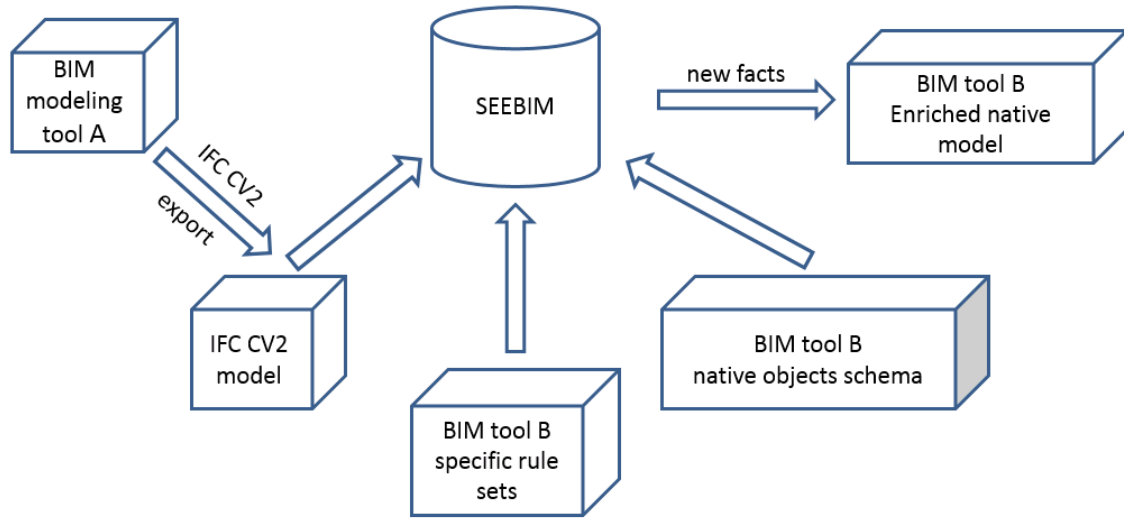
### 3 PROPOSED APPROACH

Semantic enrichment of building models is a process in which an expert system inference rule engine applies domain-specific rule sets to identify new facts about building objects and relationships in an input building model and adds them to the model. The inference rules encapsulate the knowledge of domain experts who are skilled at identifying objects and their function and behavior by observing them in the context of a building model. The semantics of a building object are composed of its form, function and behavior (Lee, Sacks and Eastman 2006). These are manifested by its shape (3D geometry), material and mechanical properties, its functional classification, its topological and aggregation relationships with other objects, that have particular meaning in a domain's context. The relative locations of objects to one another are key determinants for their functional classification and for determining their topological and aggregation relationships.

The acronym for the proposed system, 'SEEBIM' (Semantic Enrichment Engine for Building Information Modeling), reflects the notion that domain experts (architects, engineers, etc.) can 'see' information in a model that is implicit as well as the information that is carried explicitly. The rules are defined as *IF-THEN* rules using a predefined set of object types and operators, expressed in a format easily comprehensible to domain experts who are not programmers. The operators include functions for reading the existing building model, testing for geometrical and topological relationships, and for creating new objects, properties and relationships. The new and the enriched objects conform to the definitions of a receiving application or of an MVD defined for the given sub-domain. Fig. 1 describes one possible configuration of the semantic enrichment information workflow. In this configuration, a BIM tool exports an IFC file that complies, as a minimum, with the BuildingSmart co-ordination view V2.0. The IFC file is enriched according to a domain MVD, and then imported by a receiving BIM tool that has an import function tailored to that MVD. In an alternative configuration, shown in Figure 2, the SEEBIM engine can be incorporated directly within the receiving BIM tool and output an enriched model directly in the native format of the tool. In this case, the rule-sets are specific to the particular BIM tool rather than simply domain-specific. In all cases, the principal advantage lies in the fact that the IFC export file is generic.



**Fig. 1.** Semantic enrichment information workflow for IFC import.



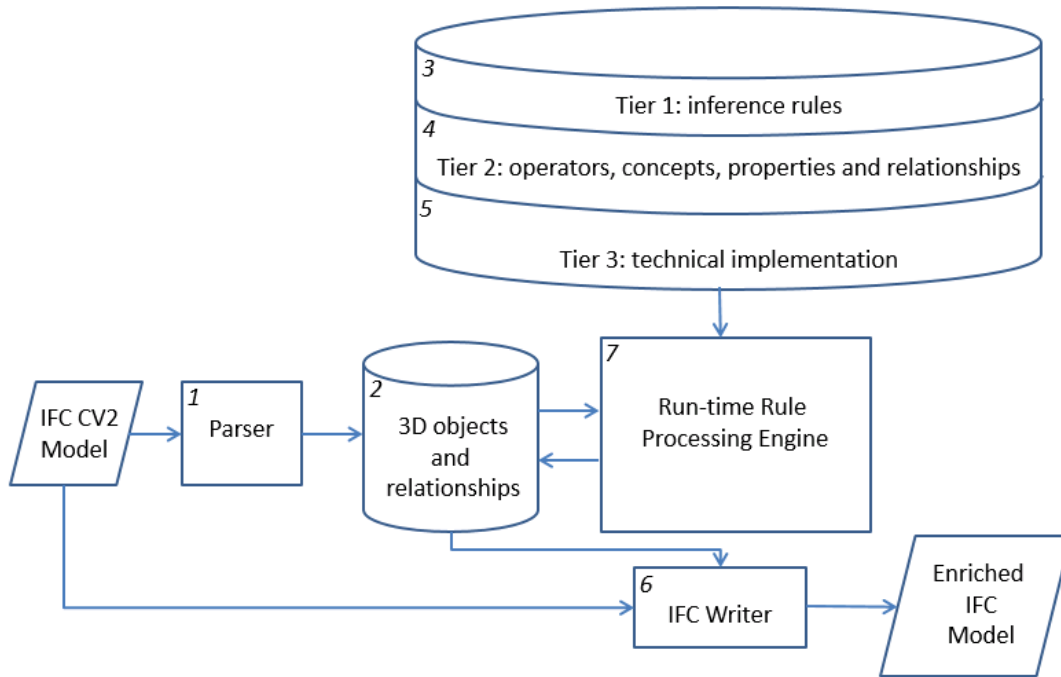
**Fig. 2.** Semantic enrichment information workflow for direct native import.

### 3.1 Semantic enrichment engine architecture

The semantic enrichment engine (SEE) utilizes forward chaining to infer new facts about a model. Its architecture is shown schematically in Fig. 3. It consists of the following parts:

1. A parser, labeled 1 in Fig. 3, reads IFC model instance files exported from BIM tools.
2. An internal run-time database, labeled 2, stores parsed objects, relationships and their attribute values.
3. The inference rules are compiled by domain expert users and stored in persistent file storage. They are compiled in a format close to natural language that uses the product model schema objects, fixed logical and relational operators, and a generic set of geometry, data query and spatial topology operators. The rule-sets are described using a three-tiered structure, as follows:
  - a. Tier 1, labeled 3 in Fig. 3, consists of the rule statements. Their lexical components are logical and relational operators, Boolean constants, universal operators (defined in Tier 2), domain specific concepts and relationships and product model schema entities.
  - b. Tier 2, labeled 4, is a library of concepts, properties, and relationships, as well as the geometry, data query and spatial topology operators that are used for compiling rules in Tier 1.
  - c. Tier 3, labeled 5, represents the implementation in computer code of the tier 2 operators.
4. An IFC writer (labeled 6) generates IFC files from the semantically enriched models in the database.
5. A custom-built run-time rule-processing engine (labeled 7 in Fig. 3) executes the user-defined rules and adds new facts about a model to the internal database of SEE. The rule processor uses forward chaining, so that derivation of any new fact about a model can trigger further new inferences. Processing continues until no further facts can be inferred.

The next three sub-sections outline the three ‘tiers’ of the rule sets that operationalize the inference of topological, geometric, spatial, aggregation and other relationships among objects for the purpose of semantic enrichment. They introduce both the universal and the domain-specific concepts and operators on model objects as well as the technical implementation aspects that illustrate the proposed approach.



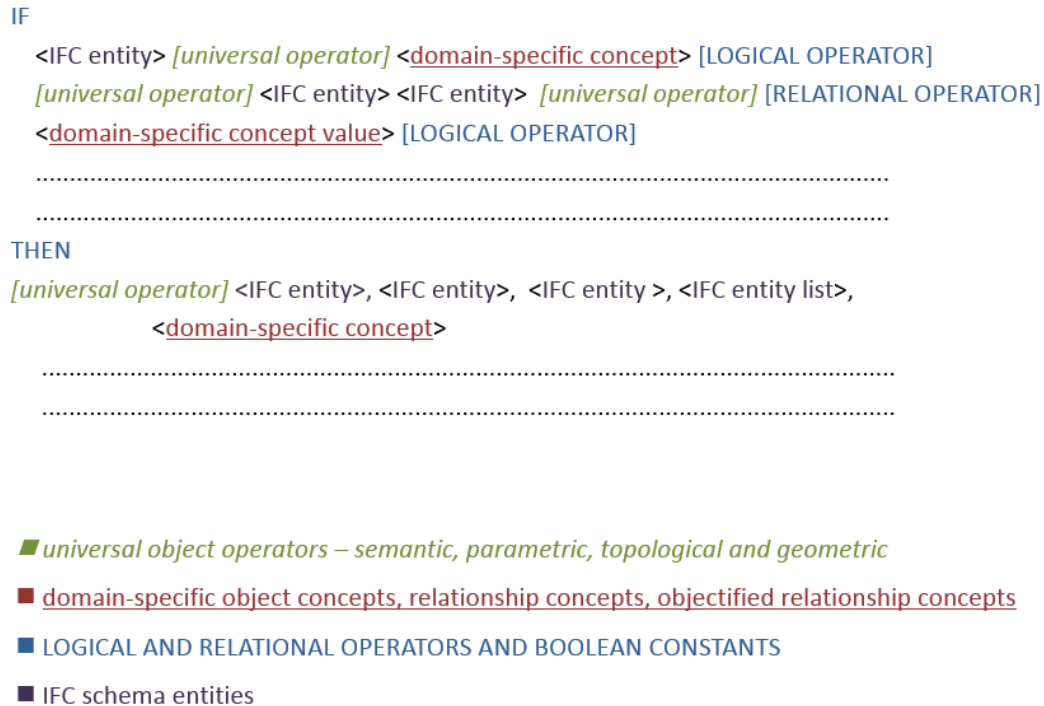
**Fig. 3.** Semantic enrichment engine architecture.

### 3.2 Tier 1

At this level domain experts can define inference rules for finding new facts about a model. Rules are collected in domain- or tool-specific sets that are used independently of one another, although rules may be copied from one set to another to make compilation efficient. The rules are processed in the run-time inference engine, which utilizes forward chaining. As shown in Fig. 4, the inference rules use universal operators and domain-specific concepts, properties and relationships, to manipulate model objects (IFC entities). The operators are defined in Tier 2. An example of a set of instantiated Tier 1 rules is provided in section 3 below.

### 3.3 Tier 2

This level provides a library of concepts, properties, and relationships, as well as geometry and spatial topology operators, which can be used to compile rules. Concepts, properties, and relationships are either universal or domain-specific. The universal concepts, properties, and relationships are universal within the universe of discourse, which in this context is the broad AEC domain as defined by the IFC Coordination View 2.0. The domain-specific concepts, properties, and relationships are defined by MVDs, such as the Precast Concrete Institute (PCI) MVD. Operators are mostly universal, but they can also be domain specific. In some cases, domain-specific operators may specialize universal operators.



**Fig. 4.** An inference rule template.

### 3.4 Concepts

- *Building elements*: These are the basic physical parts of buildings that are defined either in the IFC CV 2.0 or in the domain-specific MVD.
- *Objectified relationships*: A precast connection is an example of a domain-specific objectified relationship. The relationship object may or may not have a specific geometry, and it will usually have relationships with the objects it relates and may represent an assembly of its parts (such as steel plate embeds in a precast connection). These are defined in domain-specific MVDs.

#### 2.3.2 Properties and Relationships

- *Function*: The function of building objects is defined by their classification (e.g. *IfcBeam*). The classification in any given exported IFC file is often unreliable, because the BIM tool's export may not match the intended classification that can be context specific. For example, a steel bearing plate may be exported as an *IfcBeam* if the tool's internal representation does not model bearing plates. In some cases, users may have used massing tools to generate geometry, which are then generic objects with no specifically declared function.
- *Geometry*: Each tangible entity has a geometric representation, which can be attribute driven (parametric, usually with swept solid geometry), boundary representation (B-rep) or constructive solid (CSG) geometry. Entity properties and their values are also provided in the IFC files.
- *Material*: Building objects that have a physical manifestation are made of a material. The material property can be used for building objects classification. For example, an entity that represents in IFC model a concrete wall must have a 'concrete' in its property set description.
- *Identity*: Building objects can have different identities in different domains. For example, architectural column versus structural column. The *is\_a* operator is used for checking an object identity.



- *Aggregation relationships* relate the parts of assemblies. The relationships are defined in the relevant MVD. For example, hollow-core planks are parts of a precast slab. The *is\_related\_to* operator is used for checking if a given relationship relates two objects.
- *Composition relationships*: An object may be a part of another object. This implies that the geometry of the part is subsumed in the geometry of the whole. The *is\_part\_of* operator implements this check.

### 2.3.3 Geometry and Spatial Orientation Operators

- *Face* operators derive parts of the external envelope of objects. For example, *vertical\_narrow\_face* and *vertical\_wide\_face* are operators that returns particular faces of an object based on its geometry, proportion and orientation.
- *Orientation* in a coordinate system XYZ. Three orientation operators determine whether an axis of an object or the plane of a face is vertical, horizontal or inclined.
- *Proportion* as a shape characteristic. Building objects are commonly perceived as: *1D* - one dimension is much larger than the other two, e.g. beams and columns; *2D* - one dimension is much smaller than the other two (e.g. walls and slabs); or *3D* - all dimensions are relatively of the same size (such as pad foundations).

### 2.3.4 Spatial topology operators

These operators test the relative locations of two objects and return Boolean constants *true* or *false*.

- *Adjacency* – two objects are adjacent if they have a common face or if one of each of their faces is less than some given tolerance distance from the other. For example, *is\_adjacent\_to(tolerance)* operator returns *true* if the distance between two given objects is no more than *tolerance*. Otherwise, it returns *false*.
- *Contact* – a special case of adjacency where the objects must be in contact, i.e. the tolerance distance is zero.
- *Containment* – one object is completely contained within another spatial object. For example, the *is\_contained* operator returns *true* if one object is completely contained in volume of another object. Otherwise, it returns *false*.
- *Overlapping* – two objects have an overlapping volume (the rest of their volumes are separate);
- *Volumetric* operators derive a volume of space that is occupied by an object or a set of objects that results from the relative locations of two or more objects. For example, the volume between two objects is derived by the *proximate\_volume* operator. This operator is used by the *is\_in\_between* operator that returns a list of objects which are contained in or overlapping with the proximate volume between two given objects.

The spatial topology operators are needed because explicit topological relationships among entities in an IFC model, if represented at all, are potentially misleading and they do not necessarily reflect the topology of building elements because they are dependent on the ways in which people model. For example, a file that erroneously relates IfcColumn entities on the first and fifth floors to the third floor IfcBuildingStorey will pass IFC CV 2.0 certification. In many cases, the relationships between building storey and object are not one-to-many, but many-to-many; a precast column can often span more than one storey. For this reason, the IFC object relationship tree of a model is not a reliable source for inferring spatial or functional relationships and attributes. Assemblies, where they exist in the IFC relationships tree, can be useful, mostly for reinforcing (rebar meshes and cages). As a result, determining spatial relationships, whatever they are, is the starting point for semantic enrichment of an IFC model.

### 2.3.5 Auxiliary operators

These operators process IFC entities and their attributes, comparing and creating new IFC entities. For example, the *create\_relationship* operator creates new relationship and adds it the internal database of SEEBIM.

## 3.5 Tier 3

Tier 3 provides technical implementation of the Tier 2 concepts, properties and relationships. Many of the operators can be implemented using the tools provided by spatial query languages and function libraries. For example, a spatial extension of Oracle 9i (Oracle n.d.), which is a sophisticated widespread commercial database system, provides operators for selecting objects within a given distance of an object, nearest neighbor objects, or objects with topological relationships. The 3D ACIS Modeler (Spatial Corporation 2014) is another example of a geometric modeling kernel that provides a rich set of geometric operators.

## 4 PROOF OF CONCEPT

In order to illustrate the concept, a prototype SEEBIM application was implemented, a set of rules were compiled for the domain of precast concrete, and a series of tests were performed in which the system was tasked with identifying joints between precast walls and aggregations of hollow-core slabs. The precast concrete domain-specific concepts were taken from the precast concrete Model View Definition (MVD) (C. Eastman, R. Sacks, et al. 2010) as defined by the US National BIM Standard (NIBS 2014). The sections of this chapter describe the software implementation and the tests.

In precast concrete construction, buildings are composed of distinct pieces. The pieces must be connected structurally and the seams between walls and other pieces must be sealed with joint material. The business process maps defined in the Precast Concrete Information Delivery Manual, part of the Precast National BIM Standard, require exchange of building models between architects, engineers, fabricators and erection contractors. The MVD in the same standard defines concepts for structural connections, joints, hollow-core and double-tee slab aggregations, rebar patterns and cages, etc. It details how these concepts should be bound (implemented) using IFC entities and relationships.

To date, no BIM software tools can export precast MVD compliant IFC files. As envisioned in the SEEBIM approach, the alternative to programming precast MVD compliant exports in each of multiple BIM authoring tools is to compile a set of SEEBIM inference rules, process them to identify the precast concepts in standard IFC CV 2.0 files, and export IFC files that fully express all of the relevant precast concepts as defined in the MVD. This is the challenge adopted for the tests for proof of concept for SEEBIM presented here. Tests were conducted for precast joints and for slab aggregations. The identity, geometry and material properties of entities in IFC files generated by Tekla Structures 19 and Autodesk REVIT 2013 were used by SEEBIM to classify building elements such as beams, columns, walls, slabs, steel reinforcing and connecting hardware and to infer the joints and aggregations. In some cases, the elements' classification was context specific.

### 4.1 Prototype Implementation

A prototype 'Semantic Enrichment Engine for Building Information Models (SEEBIM)' was implemented in C# programming language in the Microsoft Visual Studio integrated development environment. The IFC parser component uses the STEP Toolbox (RDF 2014). When parsing a model, the parser saves all building objects that have geometry manifestation and their attribute values and relationships in the internal database. The input model is assumed to be exported from a BIM modeling tool according to the IFC CV 2.0. In the case of the tests described below, this means that the precast elements and connecting hardware are explicitly represented as IFC building entities with distinct 3D geometry, while concepts for precast joints and slab aggregations are not represented at all. The SEEBIM IFC file writer component creates a copy of the original IFC model and adds the newly inferred facts from the internal database – new objects, property values and relationships – to it. A user interface, implemented in an open source, server-side, HTML embedded scripting language (PHP), allows users

to define *IF* clauses of rules in a form close to natural language. *THEN* clauses of rules require a knowledge of underlying product model schema (for example, the IFC schema). The system stores user defined rules in a unique clear text file for each user account.

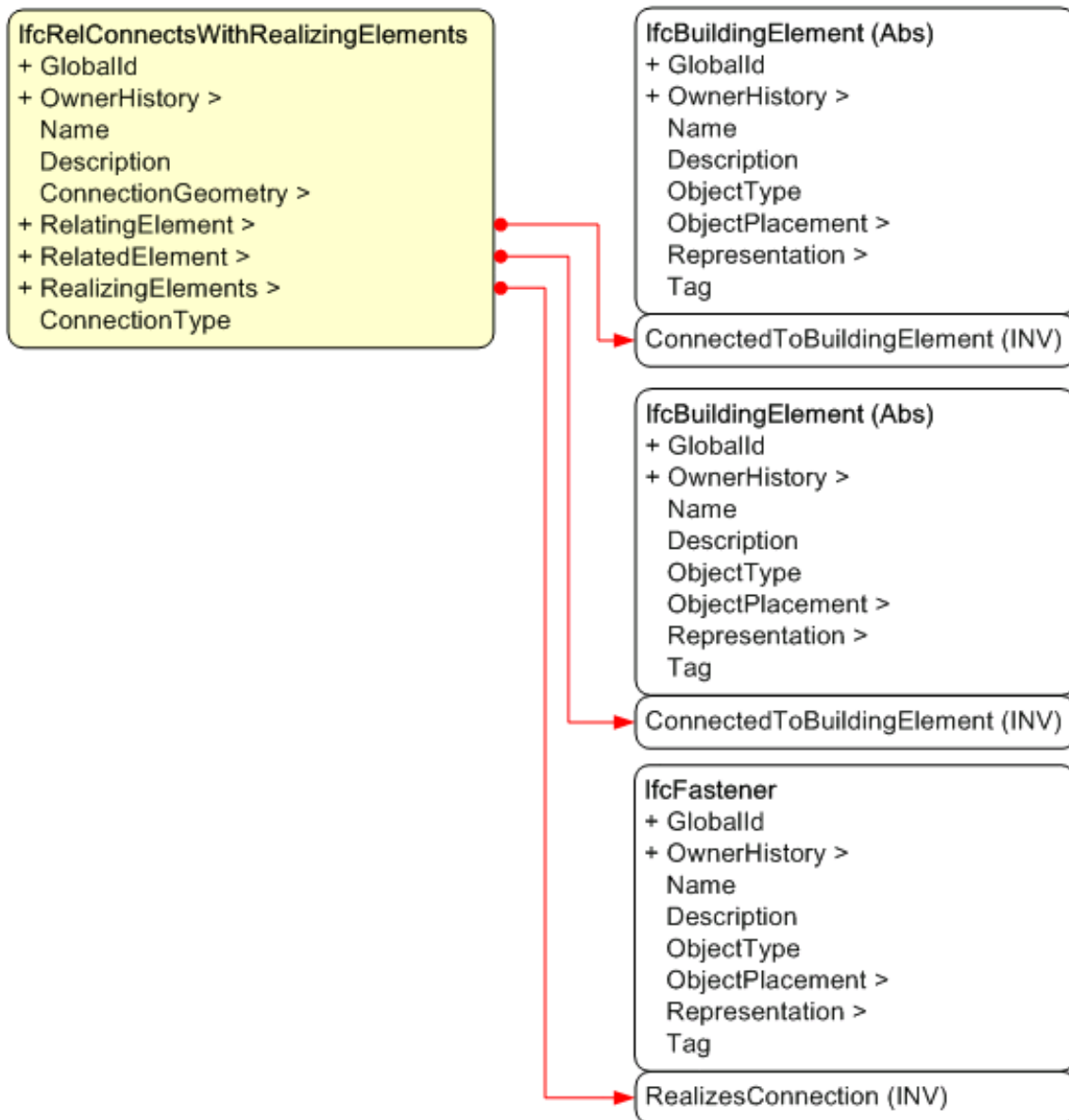
A limitation of the prototype implementation is that, for the sake of simplicity and clarity, it is restricted to an aligned bounding box representation of the building objects' geometry. The operator functions were programmed subject to this limitation. This suffices for proof of concept testing. A full implementation would require use of geometry operators from industry standard 3D geometry object libraries, such as ACIS (Spatial Corporation 2014), that can operate on complete and unrestricted BREP or CSG geometry.

The inference engine itself was developed from scratch rather than using one of many available open-source rule-processing engines, because all of the available forward-chaining rule processing engines that could be found reason using tests of facts that depend on objects of a single data type; they cannot process facts that depend on spatial or other relationships between more than one object types simultaneously. The spatial topology of building elements is defined by their relative location vis-a-vis one another, so that more than one object data type is often needed in rule clauses. The same is true for other object relationships.

## 4.2 Precast Joint Tests

In the precast concrete wall to wall joint IFC models shown below in this section, there are two wall panels and sealing strips between them. Tekla Structures 19 and Revit 2013 export the wall panels as *IfcWallStandardCase* entities and the sealing strips as an *IfcColumn* entity (sealing elements were modeled as columns). The exported files contained no reference at all to the semantic construct needed to represent the precast joint. The IFC binding for the precast joint concept defined in the PCI MVD calls for use of *IfcRelConnectsWithRealizingElements* and *IfcFastener* instances, as shown in Fig. 5. The task for SEEBIM was to infer the precast joint concept and to add it to the IFC file. Fig. 6 shows one rule from the set of Tier 1 rules capable of inferring the joint. In this set of Tier 1 rules, all of the domain specific concepts are defined in the PCI MVD and all of the universal AEC object concepts are IFC CV 2.0 concepts. Some of the rules in the set (not shown) classify the objects. Classification of an object as a 'Precast Concrete Wall', for example, requires a Tier 1 rule that considers the IFC schema functional classification, shape, proportions, material and domain-specific typical dimensions of the object. To avoid generating duplicate joints, the *is\_not\_related\_to* operator checks if two objects are not already related to each other through a 'Precast Joint' semantic construct. The *is\_adjacent\_to* operator checks whether the distance between two objects is no more than '60' mm. The *is\_in\_between* operator finds all of the objects in the model that are contained in or overlapping with the proximate volume between the two precast walls. The operator returns *true* and *objects\_list* if such objects exist. If all operators in the *IF* clause of the rule return *true* then new facts about the model are found and added to the internal database of SEEBIM. For example, the *change\_elements\_type* operator changes a type of IFC entities. In the context of the given example, type of *IfcColumn* entities, which represent sealing elements, is changed to *IfcFastener* type (see Figure 5). The *ObjectType* attribute value of *IfcFastener* entities is set to 'Sealing Strip' value. The operator *create\_relationship* adds a new *IfcRelConnectsWithRealizingElements* relationship to the database. The relationship has *RelatingElement* and *RelatedElement* attributes values equal to *object1* and *object2* correspondingly, *RealizingElements* attribute value equals to *objects\_list* that is a list of 'IfcFastener' entities and *ConnectionType* attribute value equals to 'Precast Joint'. The statements in the *THEN* clause of the rule implement the semantic enrichment itself (see Figure 7). Figure 8 summarizes results of executing the set of Tier 1 inference rules for the precast joint concept on a collection of models that were used to test the set. For models 1-3, the rule identified the precast joint and added appropriate semantic constructs to the models. For model 4, the rule did not identify a joint because the distance between the walls is greater than the tolerance for adjacency. No joint was identified in model 5, because the sealing elements were not overlapping with or contained in the proximate volume between the two walls (the proximate volume is null). However, for model 6, the rule failed. It identified the joint and updated the model accordingly, because when checking adjacency it did not consider which faces of the walls should be adjacent. Therefore, the rule was rewritten, as shown in Figure 9, to use face operators. The resulting rule gave the correct results for all cases.

## Instantiation diagram



**Fig. 5.** PCI-147: IFC instantiation binding diagram for precast joint element assignment.

```

IF
  <object1> is_a 'Precast Concrete Wall' AND
  <object2> is_a 'Precast Concrete Wall' AND
  <object1> is_not_related_to <object2>, 'Precast Joint' AND
  <object1> is_adjacent_to <object2>, '60' AND
  <objects_list> is_in_between <object1>, <object2>
THEN
  change_elements_type 'IfcFastener', <objects_list>, 'ObjectType', 'Sealing Strip'
  create_relationship 'IfcRealConnectsWithRealizingElements',
    <object1>, <object2>, <objects_list>, 'ConnectionType', 'Precast Joint'

```

- universal object operators – semantic, parametric, topological and geometric
- domain-specific object concepts, relationship concepts, objectified relationship concepts
- LOGICAL AND RELATIONAL OPERATORS AND BOOLEAN CONSTANTS
- IFC schema entities

**Fig. 6.** Tier 1 rules for inferring a precast joint concept according to PCI-147.

#### Original IFC file content:

```

#60=IFCWALLSTANDARDCASE('1HTL_90000Zp4pDZOsCpWq',#5,
'PANEL','200*3000','200*3000',#33,#59,
'ID51752f89-0000-008f-3133-363636333834');

#116=IFCWALLSTANDARDCASE('1HTL_90000Yp4pDZOsCpWq',#5,
'PANEL','200*3000','200*3000',#104,#115,
'ID51752f89-0000-008b-3133-363636333834');

#164=IFCCOLUMN('1HTL_90000XZ4pDZOsCpWq',#5,'Sealing
Strip','FLT10*100','FLT10*100',#153,#163,
'ID51752f89-0000-0086-3133-363636333834');

```

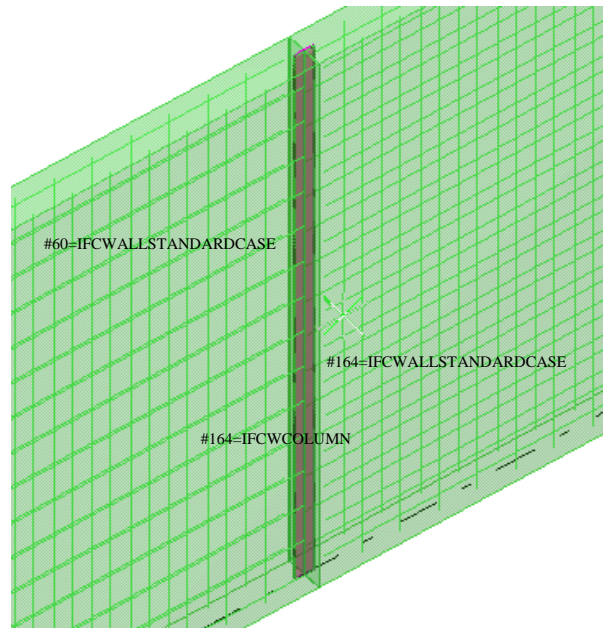
#### Semantic Enrichment:

```

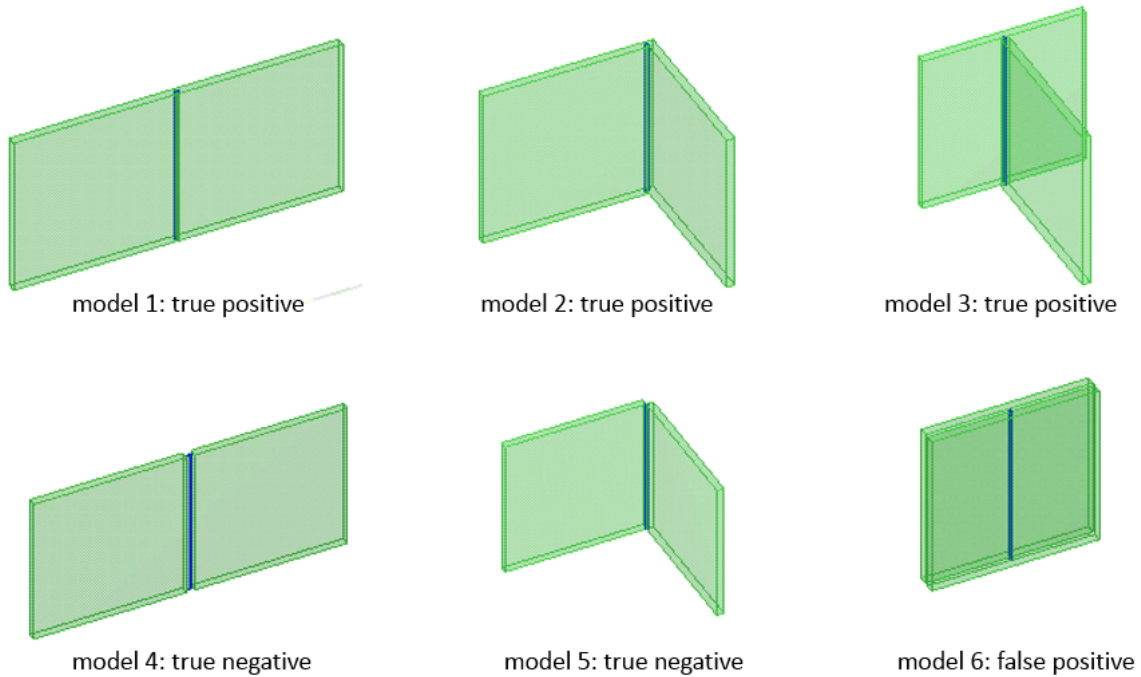
#420=IFCRELCONNECTSWITHREALIZINGELEMENTS(
'1HTL_90000Zp5pDZOsCpWq',#5,'J-1', 'Logical Joint',#,
#60,#116,(#315), 'Precast Joint')

#315=IFCFASTENER('1HTL_90000Zp6tDZOsCpWq',#5, 'Vertical
Precast Wall Panel to Wall Panel Joint Type 1', 'Precast
Joint',#153,#163,$,)

```



**Fig. 7.** An example of precast joint along vertical faces between two precast concrete walls.



**Fig. 8.** Precast concrete wall to wall joint models.

```

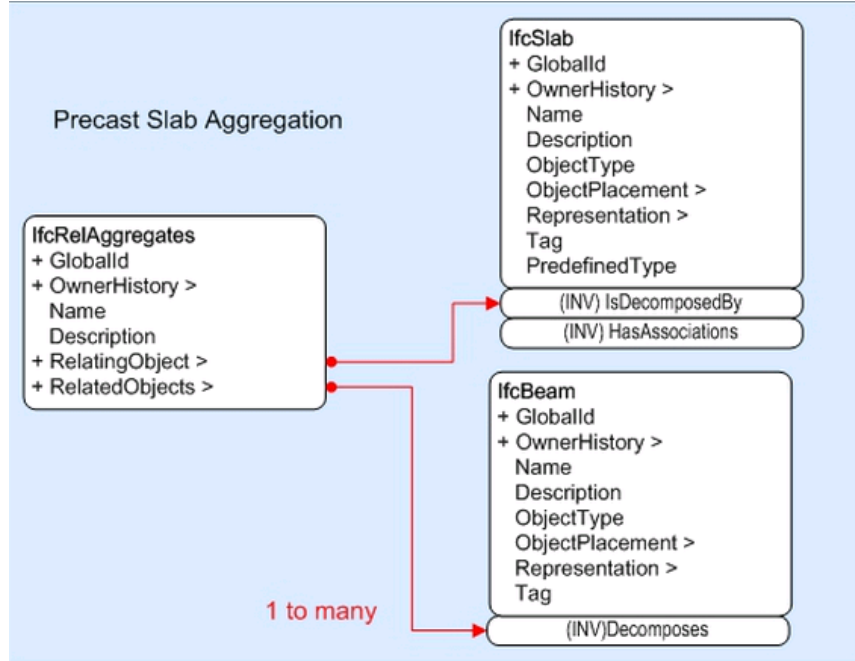
IF
  <object1> is_a 'Precast Concrete Wall' AND
  <object2> is_a 'Precast Concrete Wall' AND
  <object1> is_not_related_to <object2>, 'Precast Joint' AND
  (<object1> vertical_narrow_face is_adjacent_to <object2> vertical_narrow_face '60'
    OR
    <object1> vertical_narrow_face is_adjacent_to <object2> vertical_wide_face '60')
AND
  <objects_list> is_in_between <object1>, <object2>
THEN
  change_elements_type 'IfcFastener', <objects_list>, 'ObjectType', 'Sealing Strip'
  create_relationship 'IfcRealConnectsWithRealizingElements',
    <object1>, <object2>, <objects_list>, 'ConnectionType', 'Precast Joint'

```

**Fig. 9.** Tier 1 updated rules for inferring precast joint.

### 4.3 Precast Slab Aggregation Tests

The precast concrete slabs used for these tests are composed of hollow-core slab sections. Both Tekla Structures 19 and REVIT 2013 export the sections as *IfcBeam* entities, but the exported file contained no semantic constructs that represent the precast slab aggregations. The IFC binding for the concept of precast slab aggregation, as defined in the PCI MVD, calls for use of *IfcRelAggregates* and *IfcSlab* instances, as shown in Figure 10.



**Fig. 10.** PCI-40: Instantiation diagram for precast slab aggregation.

Therefore, the task for SEEBIM was to infer the precast slab aggregation concept and to add it to the IFC file. Figure 11 shows a single rule from the set of Tier 1 rules used to infer the slab aggregations. As in the joint example, all of the domain specific and universal object concepts are defined in the PCI MVD and IFC CV 2.0 respectively. The *vertical\_wide\_face* operator returns appropriate parts of an object's external spatial envelope. The *is\_adjacent* operator, in this case, checks whether the distance between two faces that belong to different objects is no more than '30' mm. The *proximate\_volume* operator calculates the proximate volume between two given spatial objects. The volume, being itself a spatial object, has geometry properties such as length. The *length* operator returns a numerical value of this attribute. To avoid generating multiple slab aggregations where one has already been identified (which could occur due to the iterative nature of the rule-processing engine), the *is\_part\_of* and *is\_not\_part\_of\_a* operators check whether an object is already a part of 'Precast Slab Aggregation' or not. If the object is a part of 'Precast Slab Aggregation' construct, the operator returns a corresponding IFC relationship (*object4*). In the context of the given example, it is an *IfcRelAggregates* entity. If all operators in the *IF* clause of the rule return *true* value then new facts about the model are found and added to the internal database of SEEBIM. The *create\_element* operator creates a new IFC entity (*object3*) and adds it to the database. In the given example, it is an *IfcSlab* entity with *ObjectType* attribute value equal to 'Precast Slab Aggregation'. The operator *create\_relationship* creates a new *IfcRelAggregates* relationship and adds it to the database. The new relationship has *RelatingObject* and *RelatedObjects* attributes values equal to *object3* and to a list of objects: *object1* and *object2*, and *Name* attribute value equal to 'Precast Slab Aggregation'. The *add\_object\_to\_relationship* operator adds an object to a relationship. In the given example, the operator adds *object1* to the *RelatedObjects* attribute of *object3* that is 'Precast Slab Aggregation' construct represented by *IfcRelAggregates* entity. Similarly, the *add\_objects\_to\_relationship* operator adds a list of objects (*object5*) to the *RelatedObjects* attribute of the relationship (*object3*). The *get\_relating\_object* operator retrieves an object from *RelatingObject* attribute of a relationship and *get\_related\_objects* operator retrieves a list of objects from *RelatedObjects* attribute of a relationship. The *delete\_entity* operator deletes an IFC entity.

Figure 12 summarizes results of executing the Tier 1 inference rules set for the precast slab aggregation concept on the set of models. For all models the rule identifies the precast slab aggregation and adds appropriate



semantic constructs to the model. However, for the models 3 and 4, the rule recognizes that one of the hollow core slabs is not a part of the aggregation. The rule fails in checking adjacency of objects.

```

IF
  <object1> is_a 'Precast Hollow Core Slab' AND
  <object2> is_a 'Precast Hollow Core Slab' AND
  <object1> vertical_wide_face is_adjacent_to <object2> vertical_wide_face, '30' AND
    proximate_volume <object1>, <object2> length >= '1000' AND
  <object1> is_not_part_of_a 'Precast Slab Aggregation' AND
  <object2> is_not_part_of_a 'Precast Slab Aggregation'
THEN
  create_element 'lfcSlab', 'ObjectType', 'Precast Slab Aggregation', <object3>
  create_relationship 'lfcRelAggregates', <object3>, <<object1>, <object2>>
    'Name', 'Precast Slab Aggregation'

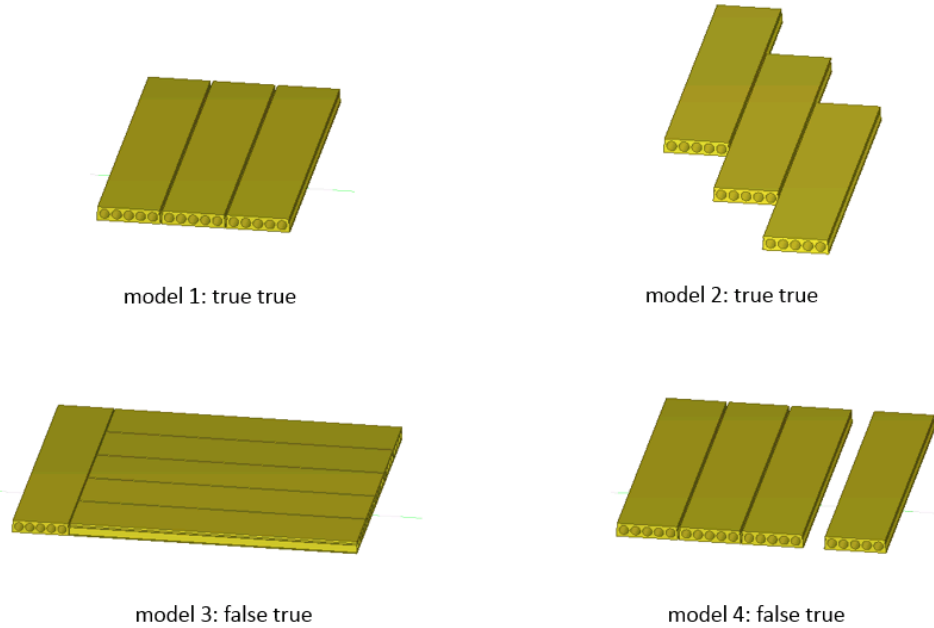
IF
  <object1> is_a 'Precast Hollow Core Slab' AND
  <object2> is_a 'Precast Hollow Core Slab' AND
  <object1> vertical_wide_face is_adjacent_to <object2> vertical_wide_face, '30' AND
    proximate_volume <object1>, <object2> length >= '1000' AND
  <object1> is_not_part_of_a 'Precast Slab Aggregation' AND
  <object2> is_part_of 'Precast Slab Aggregation', <object4>
THEN
  add_object_to_relationship <object4>, 'RelatedObjects', <object1>

IF
  <object1> is_a 'Precast Hollow Core Slab' AND
  <object2> is_a 'Precast Hollow Core Slab' AND
  <object1> vertical_wide_face is_adjacent_to <object2> vertical_wide_face, '30' AND
    proximate_volume <object1>, <object2> length >= '1000' AND
  <object1> is_part_of 'Precast Slab Aggregation', <object5> AND
  <object2> is_part_of 'Precast Slab Aggregation', <object6> AND
  <object5> is_not_equal_to <object6>
THEN
  get_related_objects <object6>, <objects_list>
  add_objects_to_relationship <object5>, 'RelatedObjects', <objects_list> ,
  get_relating_object <object6>, <object7>
  delete_entity <object6>
  delete_entity <object7>

```

**Fig. 11.** Tier 1 rules for inferring precast slab aggregation concept.





**Fig. 12.** Precast concrete slab aggregation models.

## 5 CONCLUSION

The Industry Foundation Classes (IFC) schema is widely recognized as the de-facto standard for information exchanges for BIM. IFC is a richly expressive product model schema but it lacks formal definition and is redundant (offering multiple ways to describe some object concepts and their relationships). To be useful, exchanged building model instances should contain precise expressions of domain-specific semantic constructs as required by a receiving application. As a result, in current practice, both export and import functions in BIM software tools must be tailored according to domain-specific model view definitions. The alternative approach presented in this paper simplifies the problem by placing the onus for semantic interpretation exclusively on the importing application.

Human beings perceive buildings principally as aggregations of physically tangible objects with well-defined geometry and with spatial topology relationships in specific forms as needed for each of a wide variety of design and engineering tasks. By virtue of their intelligence, their background and domain-specific knowledge, people can interpret representations of building systems and compile mental models of their components and their function. This offers the opportunity to consider an innovative geometry and topology driven approach for creating semantically useful building model files from the explicit and implicit information contained in building models exported from BIM tools. The Semantic Enrichment Engine for BIM (SEEBIM) presented in this paper is an attempt to implement such an approach. SEEBIM supplements IFC building models, exported according to the IFC CV 2, with semantic constructs defined according to the concept bindings defined in the MVD specific to the importing application's domain.

SEEBIM was implemented with a three-tiered structure of semantic enrichment inference rules and applied to semantic enrichment with examples of information exchanges specified in MVDs for the precast concrete domain defined by the US National Precast BIM Standard. These proof-of-concept level tests demonstrated how the goal of placing the onus for information exchanges on receiving applications can be achieved in practice. The tests illustrated that classification of some objects is context dependent and that some objects need rules that identify relationship concepts for their classification. The second tier library of geometry and spatial topology operators that was implemented proved to be sufficient for supplementing IFC models with the required semantic constructs

for precast joints and slab aggregations. The open-ended library of universal geometry and spatial topology object operators (Tier 2) appears to be a highly effective way to compose rule-sets, and it has the advantage that the operators can be re-used in multiple contexts. For example, the face, proportional and adjacency operators can be used to infer rebar aggregations, such as rebar patterns and rebar cages. Note that SEEBIM does not enable round-trips of building models; rather, it allows each domain application to 'see' and import the concepts it needs. In this way, models can be exchanged bi-directionally. The semantic constructs are not maintained through exchanges, they are identified from scratch in each exchange.

Applications of the approach are not limited to exchange of BIM models using IFC import functions. A software vendor could implement the SEEBIM engine within its application, defining the rule-sets needed to generate native object imports. The rule-sets could be easily extended or updated as the host application develops. SEEBIM could also be used to enrich IFC files for code checking and for MVD export validation, by preparing appropriate rule-sets. An important advantage it has over other existing tools for MVD output validation, which check only syntax and structure of the concepts, is that it can check the spatial topology of the building elements in the exported file.

The method has some practical limitations that result from its early stage of development. Thus far, although many industry and research groups are developing model views for varying aspects of the AEC information exchanges, only a few MVDs are available. The current technical implementation (Tier 3) of SEEBIM utilizes the aligned bounding box approach for representing building objects, which limits its use for real-world BIM models. This limitation can be overcome simply in full-scale implementation by using standard 3D object libraries. Finally, as with any product model exchange process, the ability of any given rule-set to produce valid semantically enriched model files must be checked. To date, although research is progressing in this area, no formalized procedure has been established.

## ACKNOWLEDGMENTS

The authors are grateful to Prof. Chuck Eastman and his team at the Digital Building Laboratory at Georgia Tech for hosting the first author during the early stages of this work, and to the EU FP7 Marie Curie Grant for its financial support of this secondment. The authors are also indebted to Peter Bonsma from RDF Ltd. for the IFC Engine DLL Toolbox and his invaluable technical support.

## REFERENCES

- Ameri, F., and D. Dutta. 2005. "Product lifecycle management: closing the knowledge loops." *Computer Aided Design & Application* Vol. 2, No. 5, 577-590.
- Babalola, O. O. 2012. *A model based framework for semantic interpretation of architectural construction drawings*. PhD Dissertation, Atlanta: Georgia Institute of Technology.
- Bazjanac, V., and A. Kiviniemi. 2007. "Reduction, simplification, translation and." University of Maribor: CIB W78, Proc. 24th. 163-1168.
- Beetz, J., J. Leeuwen, and B. Vries. 2006. "TOWARDS A TOPOLOGICAL REASONING SERVICE FOR IFC-BASED BUILDING INFORMATION MODELS IN A SEMANTIC WEB CONTEXT." *Joint International Conference on Computing and Decision Making in Civil and Building Engineering*. Montreal. 3426-3435.
- Beetz, J., J. van Leeuwen, and B. de Vries. 2009. "IfcOWL: A case of transformation EXPRESS schemas into ontologies." *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23, 89-101.
- Beetz, J., L. van Berlo, R. de Laat, and P. van den Helm. 2010. "Bimserver.org - an open source IFC model server." *CIB W78*. Cairo.
- Borrmann, A., and E. Rank. 2009. "Topological analysis of 3D building models using a spatial query language." *Advances Engineering Informatics*. Vol. 23, 370-285.
- 2006. *buildingSMART*. <http://iug.buildingsmart.org/idms>.

- BuildingSmart. n.d. *International home of openBIM*. <http://buildingsmart.com/>.
- Chen, Yuh-Min, and Ching-Ling Wei. 1997. "Computer-aided feature-based design for net shape manufacturing." *Computer Integrated Manufacturing Systems* Volume 10, Issue 2, 147-164.
- Eastman, C. M., I. Panushev, R. Sacks, M. Venugopal, S. Aram, and R. See. 2011. *A Guide for Development and Preparation of a National BIM Exchange Standard, Technical Report*. PCI, Charles Pankow Foundation, Georgia Tech, Technion.
- Eastman, C. M., R. Sacks, I. Panushev, M. Venugopal, and S. Aram. 2010. *recast/prestressed Concrete National BIM Standard documents: Model View Definitions for Precast Concrete*. Technical Report, [www.dcom.arch.gatech.edu/pcibim/](http://www.dcom.arch.gatech.edu/pcibim/).
- Eastman, C. M., Y. S. Jeong, R. Sacks, and I. Kaner. 2010. "Exchange Model and Exchange Object Concepts for Implementation of National BIM Standards." *Journal of Computing in Civil Engineering* Vol. 24 Issue 1, 25-34.
- Eastman, C., P. Teicholz, R. Sacks, and K. Liston. 2011. *BIM Handbook : A Guide to Building Information Modelling for Owner, Managers, Designers, Engineers, and Contractors*. New Jersey: John Wiley and Sons, Inc.
- Eastman, C., R. Sacks, I. Panushev, M. Venugopal, and S. Aram. 2010. "Precast/prestressed Concrete National BIM Standard Documents: Model View Definitions for Precast Concrete. Technical Report." *Precast BIM Standard Project*. Accessed March 18, 2014. [http://dcom.arch.gatech.edu/pcibim/documents/Precast\\_MVDs\\_v2.1\\_Volume\\_I.pdf](http://dcom.arch.gatech.edu/pcibim/documents/Precast_MVDs_v2.1_Volume_I.pdf).
- Eastman, C.; Jeong, Y.; Sacks, R.; Kaner, I.; 2010. "Exchange Model and Exchange Object Concepts for Implementation of National BIM Standards." *JOURNAL OF COMPUTING IN CIVIL ENGINEERING* 24(1), 25–34.
- Eurostep. n.d. *Eurostep*. Accessed May 23, 2013. <http://www.eurostep.com/>.
- Final, S., and J. Hietanen. 2006. *IFC Model View Definition Format*. Technical Report, International Alliance for Interoperability (IAI), [www.iai.international.org/software/MVD\\_060424IAI\\_IFCModelViewDefinition.pdf](http://www.iai.international.org/software/MVD_060424IAI_IFCModelViewDefinition.pdf).
- Hietanen, J., and S. Final. 2007. "IFC model view definition format." *Proceedings of the 24th CIB W78 Conference*. Maribor: International Alliance for Interoperability.
- ISO - International Organization for Standardization. 2014. *ISO 10303-11*. Accessed March 8, 2014. [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=38047](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=38047).
- ISO - International Organization for Standardization. 2014. *ISO 16739:2013*. Accessed March 8, 2014. [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=51622](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=51622).
- ISO. 2014. *ISO 10303*. April 2. Accessed April 2, 2014. <http://www.iso.org/iso/home/standards.htm>.
- Jeong, Y-S, C. M. Eastman, R. Sacks, and I. Kaner. 2009. "Benchmark tests for BIM data exchanges of precast concrete." *Automation in Construction* 18, 4 pp 469-484.
- n.d. *Jotne EPM Technology*. Accessed May 7, 2013. <http://www.epmtech.jotne.com/index.php?id=562520>.
- Lee, G., R. Sacks, and C. Eastman. 2006. "Specifying parametric building object behavior (BOB) for a building." *Automation in Construction* 15 758-776.
- Leibich, T., and J. Wix. 1998. "Highlights of the development process of industry foundation classes." *Proceedings of the Second ECPMM Conference*. Walford, UK.
- Liebich, T, Y Adachi, J Forester, J Hyvarinen, K Karstile, and J Wix. 2006. *Industry Foundation Classes IFC*. International Alliance for Interoperability.
- Lou, Y. 2011 . "A method toward building semantic-enriched product model." *ICECC*. Ningbo: Electronics, Communications and Control (ICECC). 1826 - 1830.
- Mazairac, W., and J. Beetz. 2013. "BIMQL - An open query language for building information models." *Advances Engineering Informatics* Vol. 27, 444-456.
- Mocko, G., R. J. Malak, C. J. Paredis, and R. Peak. 2004. "A KNOWLEDGE REPOSITORY FOR BEHAVIORAL MODELS IN ENGINEERING DESIGN." *ASME 2004 International Design*

- Engineering Technical Conferences and Computers and Information in Engineering Conference*. Salt Lake City: ASME. 943-952.
- Nguyen, T. H., A. A. Oloufa, and K. Nassar. 2005. "Algorithms for automated deduction of topological information." *Automation in Construction* Vol. 14, 59-70.
- NIBS. 2014. *National BIM Standard-United States™ Version 2*. National Institute of Building Sciences. Accessed March 30, 2014. <http://www.nationalbimstandard.org/>.
- Noack, R. 2001. *Converting CAD Drawings to Product Models*. Licentiate Thesis, Stockholm: Institutionen för fastigheter och byggande.
- Nour, M. 2009. "Performance of different (BIM/IFC) exchange formats within private collaborative workspace for collaborative work." *ITcon Vol. 14* 736-752.
- Noy, N. F., and D. L. McGuinness. 2001. "Ontology Development 101: A Guide to Creating Your First Ontology." Olofsson, T., C. M. Eastman, and G. Lee. 2008. "Editorial - Case studies of BIM in use." *ITcon Vol.13*, 244-245.
- Palzar, T., and Z. Turk. 2008. "Interoperability in practice: geometric data exchange using the IFC standard." *Itcon Vol. 13*. 362-380.
- RDF. 2014. Accessed February 27, 2014. <http://rdf.bg/index.php>.
- Sacks, R., I. Kaner, C. Eastman, and Y. Jeong. 2010. "The Rosewood experiment — Building information modeling and interoperability for architectural precast facades." *Automation in Construction* Volume 19, Issue 4, Pages 419-432.
- Sacks, R.; Eastman, C. M.; Lee, G.;. 2004. "Parametric 3D modeling in building construction with examples." *Automation in Construction* 13, 291-312.
- See, R. 2010. *IFC Solution Factory: Model View Definition Site*. [www.blis-project.org/IAI-MVD/](http://www.blis-project.org/IAI-MVD/).
- Shah, J. J.; Mantyla, M.;. 1995. *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications*. New York: Wiley.
- Siemens. n.d. *Synchronous Technology*. Accessed March 9, 2014. [http://www.plm.automation.siemens.com/en\\_us/plm/synchronous-technology.shtml](http://www.plm.automation.siemens.com/en_us/plm/synchronous-technology.shtml).
- Spatial Corporation. 2014. *3D ACIS Modeling*. Accessed March 18, 2014. <http://www.spatial.com/products/3d-acis-modeling>.
- Venugopal, M., C. M. Eastman, R. Sacks, I. Panushev, and V. Aram. 2010. "Engineering semantics of IFC product model views." *Proceedings of the CIB W78 2010*. Cairo, Egypt.
- Wiese, M., P. Katranuchkov, and R. Scherer. 2003. *Generalized model subset definition schema*. 20th International Conference on Applications of IT in the AEC industry.
- Won, J., G. Lee, and C. Cho. 2013. "A No-Schema Algorithm for Extracting a Partial Model from an IFC Instance Model." *Computing in Civil Engineering*.
- Zang, L., and R. Issa. 2011. "Ontology Based Partial Building Information Model Extraction." Sophia Antipolis France: CIB W78-W102 .