SEMANTIC EXCHANGE MODULES: A NEW METHOD FOR SPECIFYING AND IMPLEMENTING MODEL VIEWS

Manu Venugopal\textsuperscript{1}
Chuck Eastman\textsuperscript{2}
Rafael Sacks\textsuperscript{3}
Ivan Panushev\textsuperscript{2}
Shiva Aram\textsuperscript{2}

\textsuperscript{1}School of Civil & Environmental Engineering, Georgia Tech
\textsuperscript{2}College of Architecture, Georgia Tech
\textsuperscript{3}Civil and Environmental Engineering, Technion, Israel Institute of Technology

Report from the Digital Building Laboratory
Georgia Institute of Technology
draft 0.8
July 18, 2011
0 Overview

The Industry Foundation Classes Schema (IFC) has become recognized as the de facto standard for interoperability in the AEC/FM industry. IFC is a rich yet redundant schema providing multiple ways to represent information. The ‘Model View’ approach is recognized as the needed answer for more precisely specifying the semantics to be used in a given exchange. The National BIM Standard (NBIMS) was developed in part to address this need.

There are many industry and research groups developing model views for varying aspects of the AECO realm, using the NBIMS. However, only a few exchanges have been defined and implemented thus far. The most visible and important exchange requirements are contractually specified handovers, such as contract documentation at bid time and operations handover when construction is complete. Model views are rigid in their contents and work well when the requirements can be fixed long ahead of time. They are too rigid for problem-solving workflows within a project, where needs can quickly vary and need to support collaboration. Experienced users note that exchanges at the collaboration level need to be adaptable and flexible. There are other limitations. The current approach to MVDs require intensive efforts on the part of users, who using their experience, must specify the content of exchanges from their heads. The needs often vary case-by-case and they must normalize these to a fixed set of contents. After the domain users specify what is needed, software vendors must code numerous model view exchanges, eventually for every domain within AEC/FM that their software supports. Methods for development of model views, while organizationally defined, are not rigorous logically.

The idea of Semantic Exchange Modules (SEM) is to provide a layer of specificity above the base level of IFC. It could be considered analogous to a high level domain language on top of a general programming language; rough examples are SQL over low level data structures and the Process Language as an extension to but simplification of java. SEMs are linkable modules that can be combined to compose exchanges at run-time, that allow re-use of export and import functions for multiple domains, and that can be tested and certified as units. We also try to make them semantically intelligible – to facilitate user selection. We explore a software engineering methodology to specify the SEM structure required for IFC implementations. The objective of this approach is to make model view specifications using IFC consistent and reusable by building them from tested and validated SEMs. Such a methodology will help reduce the development time for model view and IFC implementation, support the flexible definition and implementation of MVDs and provide better understanding of model views among users.
1 What is SEM?

A SEM is a structured, modular subset of the objects and relationships required in each one of multiple BIM exchange model definitions. It has two raisons d’être: (1) to enable BIM software companies to code import and export functions in modular fashion, such that a function written to export or import model objects according to any given SEM can be tested and certified once, and then re-used to fulfill multiple exchange model exports/imports without modification; (2) to provide a common high-level specification structure that allows non-programmers to compose an MVD at run-time by defining it in terms of SEMs, allowing multiple heterogeneous platform users to specify an SEM and to facilitate automatic compilation of the MVD on both sides to the exchange.

A SEM can be defined as a binding to a set of IFC entities, attributes, relations, and functions and a corresponding set of Native model structures that carry the information associated with the IFC SEM definition. The SEM also carries the functions (methods) needed to reliably map data between the Native and IFC structures and other methods to integrate the two structures with associated SEMs.

In implementation, a SEM is a specification of a collection of variable or adapter concepts, which are the basic building blocks of MVDs. The concepts provide the details of the bindings to IFC entities, attributes and relationships. SEMs are composites of concepts, and offer correspondence with the native objects in a specific software application. The scope of SEMs will become fixed in consultation with software tool developers, since they must map not only to an Exchange Model, but also to the internal object schema of the tool.

2 Why are SEM’s needed?

Semantics in engineering and design are detailed and particular; they define a mixture of partial specifications of reality, including idealizations for form and function and

---

1 Concepts were identified by the BLIS group and others as repeated IFC specification that could be re-used. They are part of the current NBIMS implementation guidelines.
behavior of that reality. Different levels of realization and different levels of function and behavior sometimes need to be represented to define building information under different conditions. For example, precast concrete beams are pre-tensioned; for some purpose should the beam be represented in its nominal idealized shape, or its elongated fabrication shape prior to pre-tensioning (which will shorten it), or in its post-fabrication cambered shape which will be its shape upon erection, or its analytical model for structural analysis? Buildings are made up of different systems that each have their own entities and behavior, as well as shared ones. IFC attempts to provide a schema to specify this range of realization and abstraction, of both designs and real buildings and their various analytical models. There needs to be multiple ways of representing information to be exchanged, for different purposes and context. While human minds are able to mentally switch between different levels of abstraction and realization at different times, software applications need clear definition of the intended semantics. IFC defines multiple entity structures that have similar but semantically different interpretations. While some of these are well-defined (different types of geometry), others are left to user determination or convention. Some of the implicit semantics are described in the IFC documentation whereas other semantics are left to the users and future work.

To overcome this situation, the level of specificity in IFC needs to be raised to match the levels used and required in construction. The National BIM Standard does this partially, by defining model views for exchange purposes. The introduction of model view ‘Concepts’ begins to modularize IFC bindings

3. Requirements for SEMs

The requirements for defining SEMS should provide clear implementation criteria, so that the requirements can be used to clearly guide their specification and development. They should help in defining the level of aggregation and semantic definition of SEMs. A specific set of criteria and scale for measurement of these requirements is being developed, reflecting discussions with implementers. There are different scales of measurement such as nominal, ordinal, interval, etc. and different types of criteria such as necessary, sufficient and desired.

A. Composability – Each SEM should be composable with no broken links with other SEMs. Specifically, a SEM should allow bindings with other SEMs, without editing their interface, or adding or subtracting of references external to the SEM. Composability allows re-usability.

B. Coverage – the Available SEMs should address all the semantic definitions now used within IFC translators and support new IFC extensions where needed. Together, they should not prohibit any semantic coverage allowed by IFC.
C. Parsimoniousness – SEMS should aggregate bindings to the largest extent possible that does not eliminate semantically meaningful options. If one binding always includes another, then they should be included in the same SEM. Some concepts, such as `IfcLocalPlacement`, and low-level geometry, like `IfcLine` and `IfcCircle`, which are used widely and are a standard placement structures for physical objects. Instead of making a separate SEM for such repeated structure, they should be embedded into the SEMs that use them.

D. Semantic Clarity – each SEM should define a distinguishable semantic construct, easily distinguished on a use basis from all others. Each SEM must have a clearly defined human readable definition that can be used for composition and application to IDM or use case requirements.

E. Correctness - Correctness is the ability of entities to satisfy the use case specification. Correctness is the prime qualifier. It ensures whether the SEM satisfies or represents what the use case in an IDM specification is. Methods of correctness are conditional and are based on testing.

F. Reusability – Reusability is the ability of SEMs to serve for implementation of many different model views. 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 model views, which is expensive and time consuming. For new MVD development, a SEM should be defined so as to address all anticipatable uses. Modification and extension leads to retesting, which is to be avoided. Such modular SEMs can be plugged in wherever there is a requirement. The implication is that a SEM should be general enough to support all its potential uses, beyond those uses initially targeted.

G. Traceability – It should be possible to trace the origin of a model view back to exchange requirement (Synonymous to reverse engineering). Model views represent different levels of detail; hence the new methodology should contribute to a better understanding of model views by providing a concise and object oriented view of the exchange. This can also be seen as verifiability and goes back to maintainability of model views.

4. Desired Features of SEM:

The desired features are a secondary set of goals that should part of the final objectives defining SEMs and helps to improve the overall model exchange process.

A. Ease of use - Ease of use is the ease with which people of various backgrounds and qualifications can learn to use SEM and apply them to solve problems. AEC industry experts should be able to define model views based on SEMs. Knowledge of IFC will not
be needed. In terms of ease of use SEM is positioned as an intermediate layer to natural language (very easy) and high level programming languages (very complex). The intended advantage is that domain experts as well as programmers can understand model views represented in terms of SEM.

B. Rigor or Formalism - Formalism is the level of standardization and consistency achieved using standard protocols. The SEM is the fundamental building block for the exchange requirement, but what should be the granularity, atomicity, etc. of these modules? A first step would be to make explicit the background meaning about the IFC entities and relationships that are currently implicit. Formal approaches can also reduce the load on testing by introducing assertions and constraints and helping in debugging.

C. Extensibility – Ease of adapting modules to changes of specification. We need extensibility because IFC is an extensible schema and new requirements for various domains are identified and proposed in over time. By following a simple and decentralized approach it is easier to adapt to changes. Also, the more autonomous the modules, the easier it is to introduce changes.

D. Cycle time: The current model view development lifecycle of 2-3 years should be reduced to a more practical days and weeks. This will help to introduce IFC implementations in a timely manner.

5. Semantic Exchange Module Specification

5.1 Two Dimensional Graph Structure
The notion of a SEM is that it is a subset of a product model schema that can be used to create various, higher-level, model view definitions (MVD). A SEM graph (Figure 2), characterizes them in two dimensions. The first dimension is the classification hierarchy of different involved entities and relationships. The second dimension involves the implementation of each of these nodes in the graph by mapping it to a schema (IFC and native). The branches of this dimension represent the data access paths. Therefore, a SEM has: (i) a definite mapping to a schema, (ii) when fully defined, also mappings to a native model, (iii) methods to map between the two bindings, (iv) access the data and (v) belongs in a specific classification hierarchy. Such a structure allows SEMs to be executable.

Figure 3  SEM structure showing precast piece with a semantically determined geometry

The main criteria to be satisfied for creating such executable modules is composability as explained in previous section (and figure 3). Can we produce model views by carefull combining SEMs with each other?

SEM are not autonomous/independent from each other. Thus there is some need for functions to define relations between SEMs, especially those organized hierarchically. For example, if we have such exchange modules for B-Rep geometry, placement, material, features, etc., then it should be possible to compose them together to satisfy a building model view. Geometry and placement, however, has to be embedded in the
spatial configuration hierarchy. This is analogous to building a system from standard predesigned elements, where one type of system supports others. While composability can be considered a bottom-up approach, this is in clear contradiction of how IFC is designed (Top-down structure). See the next section for a deeper investigation.

5.2 Independent EXPRESS Subschema

Another main criterion of SEM is that they need to be stand-alone and testable from the completeness point of view. SEMs should be composable into a complete subschema that has no broken links or references. This criterion is synonymous to decomposing a complex EXPRESS schema (or a model view) into a small number of less complex, valid sub modules, connected by a simple structure. This should be independent enough to allow development to be done separately using these sub-modules.

Two criteria:

i. The dependencies between modules (SEMs) should be kept to a minimum.

ii. All dependencies should be explicitly defined.

![Figure 4 Semantic Exchange Module for Spatial Containment](image)

An example is the spatial configuration SEM shown in Figure 4. The project-site-building-building storey-space can be combined into a few modules and the dependency is the spatial containment relationship, which is used to assign an object into this configuration. In other words how other modules make use of this module should be clearly documented. Similar examples are the relation between structural members used for structural analysis and the physical incarnation of the members; these cross reference relations must also be built and maintained.
5.3 Object-Oriented Principles

Open-closed principle: Modules should be open for extension and closed for change. A module is said to be open if it is still available for extension. For example, it should be possible to extend its use to other domains by adding external entities.

A module is said to be closed, if internally the entities and relationships between them are well-defined and need not be changed for different contextual use. All SEMs are to be classified as open or closed, where ‘closed’ is an assertion of completeness.

If a SEM violates this principle then it is an indication that the module needs to either be broken down into more than one smaller modules, or maybe in some cases the module needs to be expanded to include more entities. This could be a guideline in drawing the boundaries of SEMs. A ‘closed’ SEM may be re-open-ended for undertaking new extensions not previously anticipated.

Some suggested additional guidelines:

1. If IFC entity structures are always composed in a given way, they should be combined into a SEM.

2. Conversely, SEMs should have boundaries corresponding to variations in binding structures.

3. If a structure is optional and not always used, but always has the same structure, it should be included in a single SEM, to aid simplification and parsimony. (Example is the spatial configuration hierarchy.)

4. Procedural realities sometimes require that certain operations are carried out incrementally, in response, for example, to the structure of a given model instance. Thus the complete structure of a potential SEM cannot be all defined at one time. In such cases, the incremental inputs need to be defined separately, as lower level SEMs, so they can be executed as needed. An example of the populating of the Spatial Configuration Hierarchy. While the overall structure is known and is generally deterministic, each Building and Storey are defined incrementally, as they are encountered in the model.

(Note: these variations apply to object structures (Entity and Type). Attribute-value dependences across SEMs are often necessary and need to be documented, but do not require partitioning.)
6. Implementation of model views based on SEM

Weak coupling: The interfaces between modules should be as minimum as possible. This allows modular continuity and protection. A system can be said to be continuous if a small change in the specification triggers the change of the least number of modules. Protection is useful if one of the modules needs to be redefined, then the change is restricted to only that module or to the least number of neighboring modules (encapsulation in object-oriented programming).

Design patterns: Following established OO design patterns help in reusability. (to be looked in more detail)

The process of generating a model view exchange is illustrated in the below figure 6. The exchange requirements have a direct mapping to the SEM structure (intuitive) and provide a means to develop new MVDs in a plug-and-play manner. SEMs are predefined in a library by packaging entities together as a module on a semantic basis. The process begins with the user entering the exchange model requirements in terms of SEMs.
For example, the user selects the required SEMs for her case. For example, the general definition of the precast building element type includes the options shown in figure 7. At the top level are precast doubletees are to be exchanged, with extruded geometry, precast concrete materials, placed relative to a grid, with connections omitted. The corresponding SEMs such as slab, extruded geometry, placement, aggregation, and property sets, etc. will be selected. The system will generate an EXPRESS schema based and its associates pre-defined mappings with the native data structure, ready for execution. Extensive work and time will be saved by this new method.

The capabilities of SEMs begin with the definition of a single SEM and its IFC binding, incrementally adding to the specificity of the various elements that are to compose the model view. The first level of executability is the composition of concepts and the definition of a syntactically correct IFC model view. This should be supported with a UI shell supporting directed composition showing the possible compositions of SEMs, given those already selected. It would include simple syntactic checking. The second level of executability is executing the SEM composition structure to link the corresponding IFC and native SEM structures and the linking process to provide user access and control.
7. SEM Example Definition:

SEM Example Definition:

SEM Example Definition: SEMs are defined to take Concepts to a higher level, to provide a level of IFC structures with precise semantic definitions, for both human and machine level interpretation. While currently defined MVD concepts are helpful at the implementation level, they are too fine-grained for BIM users to aggregate at run-time into actual exchanges. Many of them are indeed defined at a technical level (features of solid geometry, for example) that is inappropriate for direct use by engineers or architects. A higher-level construct is needed, that reflects their function, not in computer language terms, but rather their design or engineering use. This is provided by the SEMs.

Error! Reference source not found. shows an example of an Concept with the IFC binding definitions that relates a physical product (defined with IfcProduct) and all its subtypes with a shape (defined as links from IfcShapeRepresentation. It specifies the properties and assignments of the relating object, IfcProductDefinitionShape. The concept defines the use of a single IFC entity and specifies how its properties and relations are to be assigned.

We learned from the current concept definition effort that the structure of bindings is an directed acyclic graph, as described in Section 5. Top-down, they start with high level object definitions linking to multiple lower levels, eventually to bindings. To be composable, a SEM must deal consistently with its interfaces, both at the more aggregate object level and below it in property details. In the Geometric Representation example in Figure 2, the concept is just the yellow marked entity, IfcProductDefinitionShape and its bindings. It is defined to relate a shape to a Building Element. However, shapes can be associated with instances in two different ways: at the individual level shown here and through Element Types. These bindings are shown in Figure 9 and Figure 10.
Figure 8. Example of the IFC binding for a reinforcing bar concept, using the IfcSweptDiskSolid geometry.
Figure 9: Relation of shape representation to a single object.

On the upper side, the shape structure, structured under IfcShapeRepresentation, needs to be joinable with all elements that may be assigned shapes (composability criterion) - to both all BuildingElements and also BuildingElementTypes. IfcElement, the abstract supertype of all building element individuals, provide a direct pointer association using IfcProductDefinitionShape. However, that structure is different in IfcBuildingElementType, It uses IfcRepresentationMap to define a similar relationship. Thus both types of links are required. The ProductDefinitionShape relation needs to be created whenever a shape to element link is required; the RepresentationMap relation is used whenever a shape to element type is called for. One SEM can deal with both cases. The relation between shape and its reference can only be defined after the shape and its reference have both been defined.

Figure 10: Shape representation associated with an ObjectType.

Add link to IfcSolidModel
On the upper side of IfcShapeRepresentation, the similar kind of generalization is needed for location placement. All object individuals also have a location in the project. While Element and ElementType have a similar structure, their purpose is quite different. All element instances have a placement relative to another coordinate system, possible the project coordinate system. However ElementType assigns a placement origin for the object in its own coordinate system. It is the reference origin for instance placement. This is the reference origin of the object type. These are not user-defined distinctions but system one and should be addressed automatically. Thus they are defined automatically for Element and ElementTypes. The SEM boundaries should distinguish where user-driven distinctions comes into play.

From the point of view of software developers, an economy of scale is gained by defining SEMs as parametric compositions of concepts, for two reasons: a) they can be tested and certified as units, b) ideally, the functions written to export and import SEMs should themselves be modular and re-usable, thereby reducing the efforts required for implementing future model views. The current model view development work implies significant waste, because there is repetition in the work for different domains. Different groups generate overlapping concepts and IFC bindings based on their own requirements. For example, the same MVD concepts for reinforcing bar, rebar arrays, etc., can theoretically be used for the two domains of precast concrete and cast-in-place concrete. But this requires that the original SEM development consider all the likely applications.

One aim of SEM development is to modularize such routines and reduce the effort needed for implementing IFC translations. Moreover, if each such module is independently tested and validated, then a future model view generated need to be tested only for any new SEM additions, as any reused SEM is already validated. Hence, validation and certification costs can also be reduced.

8. SEM Implementation

Each software company will find its own best way to implement SEMS. The boundaries discussed above are defined in the context of the IFC schema. A difference boundary will exist in a native application, resulting is slightly different boundaries. The challenge is reconciling the two sets of boundaries.

Referring to Figure 1, notice that the mapping within a SEM on reading-in a model view is: reading an IFC file and use it to define the corresponding object in the host native application. There should be undertaken SEM by SEM in the rough order of the user definition, top-down. On export, the reverse mapping is created: from native structure to IFC. In both cases, however, we assume these will be top-down, so that references in lower level SEMs can be assigned to the higher level SEMS upon their definition.
A growing number of SEMS are available on the Precast BIM Standard website:
http://dcom.arch.gatech.edu/pcibim/

References:
- NBIMS and Hietanans work
- IFC kernel
- Precast NBIMS documents (IDM, MVD, Test Files)
- Testing and Validation - Richard See and IABI
- Development of computable rules - Jin Kook Lee, Nick Nisbet
- Process and Object models – Schapke 2010
- Semantics of Model Views for Information Exchanges using the Industry Foundation Class Schema, M. Venugopal, C.M. Eastman, R. Sacks, J. Teizer
SEM Documentation template:

**SEM Name**

SEMS defined here – may be one or many

**Overview**

(Major SEM design issues and intended semantics)

<table>
<thead>
<tr>
<th>SEM Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifier:</td>
</tr>
<tr>
<td>SEM Name:</td>
</tr>
<tr>
<td>Author:</td>
</tr>
<tr>
<td>Organization:</td>
</tr>
<tr>
<td>History:</td>
</tr>
<tr>
<td>Created:</td>
</tr>
<tr>
<td>Revisions:</td>
</tr>
<tr>
<td>SEM Description:</td>
</tr>
</tbody>
</table>

**Methods**

(the methods expected to be needed to create a new instance of the SEM)

**Concepts aggregated into this one:**

(the PCI Concepts embedded in this SEM)
working draft

IFC Diagram