

Fluency in the alphabet soup of 3D modeling is becoming increasingly more important in ensuring optimal interoperability.

Phantom Modeling in BIM

Sundeep Veguru
TJ DeGanyar, Ph.D., PE

Generally, buildings require multiple 3D models for their full design, engineering, construction, and integrated practice. At the core of an integrated practice lies the intimate collaboration between the design team, construction team and a digital 3D model. *Phantom Modeling in BIM* will illustrate varying collaborative approaches to understand how we can maintain “Phantom” links of shared 3D models between architects, contractors, and sub-contractors alike to incorporate design changes with greater efficiency.

INTRODUCTION

A basic tenet of American law is that construction trades cannot be held responsible for errors when they have properly followed contract documents. Known as the Spearin doctrine, this legal case serves as a safe harbor for contractors by implying a warranty from the design documents provided by the owner. This doctrine suggests that if the contractor follows design documents provided to them, there is no liability for performance failures or defects in the construction arising from the design.

Under traditional construction contracting paradigms, the construction trades are required to follow the contract documents, which commonly include 2D drawings and specifications. However, the design and construction industry has recently undergone a significant shift away from the use of two-dimensional CAD and paper drawings for three-dimensional, semantically rich, digital models.

In contrast to exchange plans – via drawing files like dxf or dwg – **the IFC exchange is strictly model based**; a wall is not a set of lines but **an object with specified attributes and relations.**

This trend has reached a point where this technology, generally referred to as Building Information Modeling (BIM), is being used by the majority of the industry. Additionally, the push for the 3D model to be used as a contract document – binding parties to the same extent as drawings or specifications always have – is currently increasing.

In a BIM design, the designer can select pre-programmed objects embedded with information about all of its relevant characteristics, not just its shape. Moreover, software such as Revit allows any change in plan view to automatically update any section affected by the change. In Tekla Structures, changes in dimension or geometry automatically update details and related documents.

The gap that exists today arises from each stakeholder maintaining his or her own 3D model. When an architect or engineer releases updates or addendums to the design documents (for example, if a column is removed to create a larger bay from one of the models), the architect releases a new set of drawings, making contractors and sub contractors responsible for manual reconciliation and model synchronization. Manual changes in the drawing sheets and re-detailing the connections increase the likelihood of human errors, with potential to “break” the existing model. As projects grow and file structures get more complex, this style of coordination also becomes increasingly complex, making traditional project management methods ineffective when exchanges need to be processed rapidly and accurately.

This paper provides preliminary ideas for how to use BIM/Industry Foundation Class IFC 2x3 and/or XML based IFC file exchanges to bridge the gap by creating phantom links between an architect’s model and user’s 3D phantom model, which will update all rapid changes without human-error; thus creating “phantom modeling”.

OBJECTIVE OF THE IFC

Efforts throughout the past decade have been directed at using a central repository or model server that acts as a base for interoperability between various architecture, engineer and construction disciplines and their software applications. Recently an international standard for information exchange of BIM data – the industrial foundation class (IFC) – was developed by the International Alliance of Interoperability to facilitate interoperability in the building industry [3]. The goal of IFC is to enable interoperability between building information systems.

The IFC provides a specification of a data model that covers the domain of building information. It can be used as a shared data model or integrated data base by many groups, allowing any participant involved with the planning and construction process to use the same model, thus increasing transparency of changes and the ability to let other players know of the actual change in the planning. In contrast to exchange plans – via drawing files like dxf or dwg – the IFC exchange is strictly model based; a wall is not a set of lines but an object with specified attributes and relations.

IFC 2X3 FINAL
ARCHITECTURE
SHORT FORM DISTRIBUTION

■ IFC2X2 PLATFORM
- IFC 2X PART EQUAL
TO ISC/PAS 16739

■ NON-PLATFORM PART

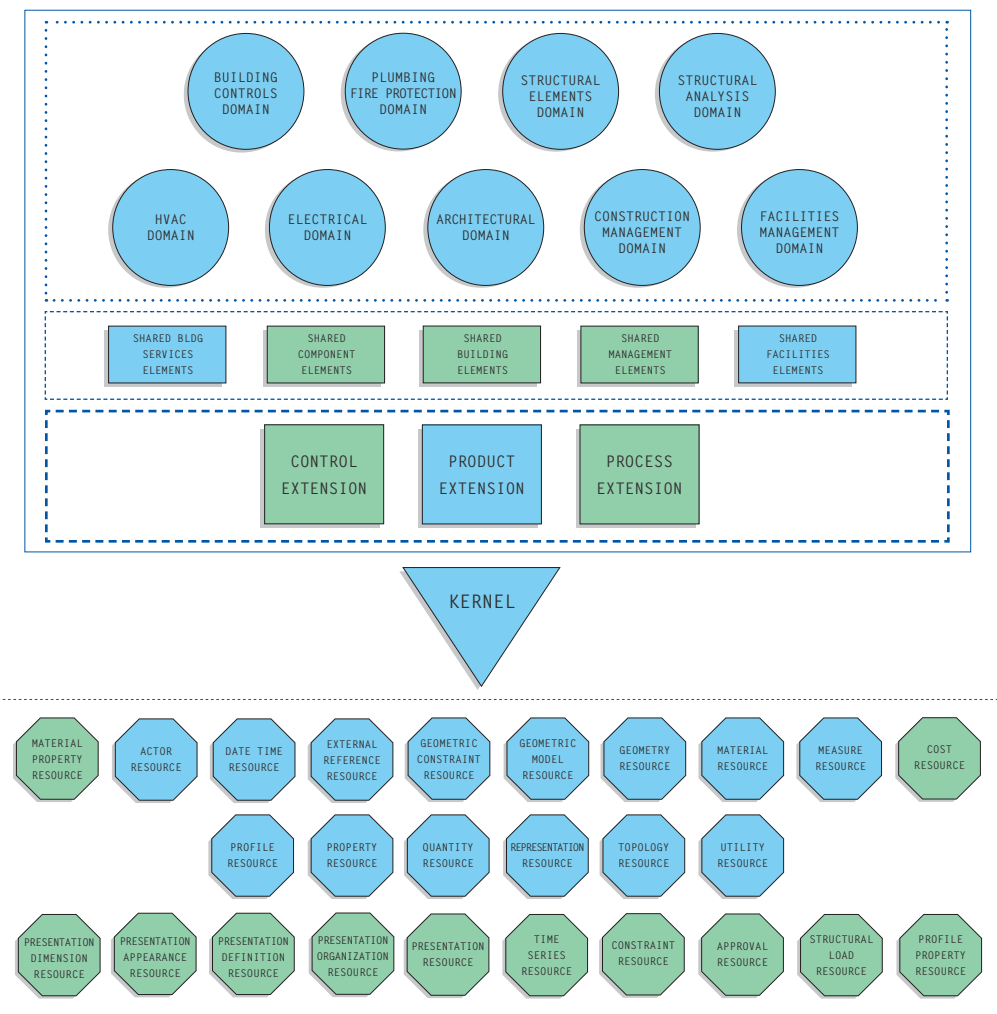


Figure 1: IFC Architecture adapted from www.iai-international.org.

ARCHITECTURE OF THE IFC

There are four layers in the IFC Model. The layers follow the “gravitation” concept, where elements of a certain layer can only refer to entities of the same or a lower layer.

- 1. Resource Layer.** This layer contains the fundamental concepts expressed as entity types such as geometry (point, line and curve) topology (vertex, edge, face and shell), geometric model (CSG, B-Rep, and Geometric Set). The elements of this layer can be referenced by elements of all other layers. In figure 1, the resource layer is symbolized with octagons.
- 2. Core Layer.** This layer declares abstract concepts that are specialized by the layer above. There are abstract concepts like

object, group, process, property definition, relationship or root. There is no instance of an abstract entity type. An abstract class provides an interface to the derived (specialized) entity types. In figure 1, the core layer is symbolized with triangle and rectangles.

3. Interoperability Layer. This layer defines basic concepts for interoperability between different domain extensions. Shared building elements like beam, door, roof, window or ramp are defined in this layer. In figure 1, the interoperability layer is symbolized with rectangles and squares.

4. Domain Layer. The entity types of the domain layer extend the concepts of the interoperability layer. Elements of one domain are not allowed to reference ele-

ments of any other domain. Domains include architecture, facility management, electricity or structural analysis. In figure 1, the domain layer is symbolized with circles.

The latest IFC release is IFC 2x3 TC1. Version 2x3 has introduced the ifcXML specification by using XML schema to define the IFC models in parallel with EXPRESS [4]. IfcXML offers a combination of advantages, first using XML technology as the mainstream technology for information publication and exchange, whilst reusing a well established, internationally recognized and supported data standard. It is recommended for architects whose partner’s applications cannot read the original IFC format, but can manage .xml databases (such as budget, energy calculations, etc.). This is an uncompressed format, with larger file sizes than the normal IFC format.

COMMON TERMS

XML: Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML. Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.

XLINK: Defines the XML Linking Language which allows elements to be inserted into XML documents in order to create and describe links between resources. It uses XML syntax to create structures that can describe links similar to the simple unidirectional hyperlinks of today's HTML, as well as more sophisticated links.

XPATH: Notation that can be used by XML Style sheet Language Transformations (XSLT) to "flatten" the XML tree structure in an XML instance file by providing a unique path definition to data contained within an XML instance file.

XSD: XML Schema (XSD) is the schema definition language used in ifcXML to describe the structure of ifcXML files. The XSD is automatically generated from the IFC source definition, the IFC EXPRESS.

ifcXML methodology: The methodology to automatically convert the IFC source definition into the ifcXML schema and to convert data sets into ifcXML documents.

ifcXML schema: An XML schema defines the set of rules that the model should follow, such as the hierarchy of the building elements, required properties that have to be defined based on their tags, etc. One can add more rules to the schema in addition to the ones given by IAI. The most recent ifcXML schema is ifcXML for IFC2x3.

ifcXML file: An XML document that can be validated by an ifcXML schema. Each ifcXML file should be well structured and validated against a valid ifcXML schema.

EXAMPLE

Structure of a building storey element and its properties in an ifcXML file

```
<IfcBuildingStorey id="bs1"><GlobalId>3XjaeTTcDeDeeddVPFcs98</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true" href="enclos1" />
  </OwnerHistory>
  <Name>First floor</Name>
  <ObjectPlacement>
    <IfcLocalPlacement xsi:nil="true" ref="i20" />
  </ObjectPlacement>
  <CompositionType>element</CompositionType>
  <Elevation>350.0</Elevation>
</IfcBuildingStorey>

<IfcRelDefinesByProperties><GlobalId>1fvr_56SS7jsK_vrADYHZ</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true" ref="enclos1" />
  </OwnerHistory>
  <RelatedObjects ex:cType="set">
    <IfcBuildingStorey ex:pos="0" xsi:nil="true" ref="bs1" />
    <IfcBuildingStorey ex:pos="0" xsi:nil="true" ref="bs2" />
  </RelatedObjects>

  <RelatingPropertyDefinition>
    <IfcPropertySet><GlobalId>1lpBu^kBSSWM7r$$$xnE1xi</GlobalId>
    <OwnerHistory>
      <IfcOwnerHistory xsi:nil="true" ref="enclos1" />
    </OwnerHistory>
    <Name>PSet_BuildingStoreyCommon</Name>
    <HasProperties ex:cType="set">
      <IfcPropertySingleValue>
        <Name>AboveGround</Name>
        <Description> Indication whether this building storey is fully above ground (TRUE), or below ground (FALSE), or partially above and below ground (UNKNOWN) - as in sloped terrain.</Description>
      </IfcPropertySingleValue>
    </HasProperties></IfcPropertySet>
  </RelatingPropertyDefinition>

</IfcRelDefinesByProperties>
```

Figure 2. Example code from an ifcXML file.

METHODOLOGY

This section suggests a methodology for implementing phantom modeling process using ifcXML. It is not fully developed and merely acts as introductions to the examples, which are for illustration purpose only.

WORKFLOW

The process to maintain phantom links consists of six steps:

1. Determine what information needs to be read or written from architect's and other user's model(s) for synchronization.
2. Review the IFC schema to see which entity definitions are relevant.
3. Design and document a global mapping table such that there is a one-to-one mapping of all the elements between both models.
4. Write application code for "import" and/or "export" that use an XML parser, such as Xerces, and generate an ifcXML instance file; use DOM or SAX application programming interface(s) to parse the XML instance file to get or set information into the mapping file.
5. Run step 4 with the architect's model and store the global IDs of model elements in the first column.
6. Repeat step 4 with user's model and store the corresponding global IDs in the second column, which will in turn give you the Xpath to the element in the ifcXML schemas that match the data item(s) you want to compare.

Note: If architect and/or specialty contractor uses Revit, export the model to IFC 2x3 and then convert it to ifcXML (since Revit doesn't support ifcXML).

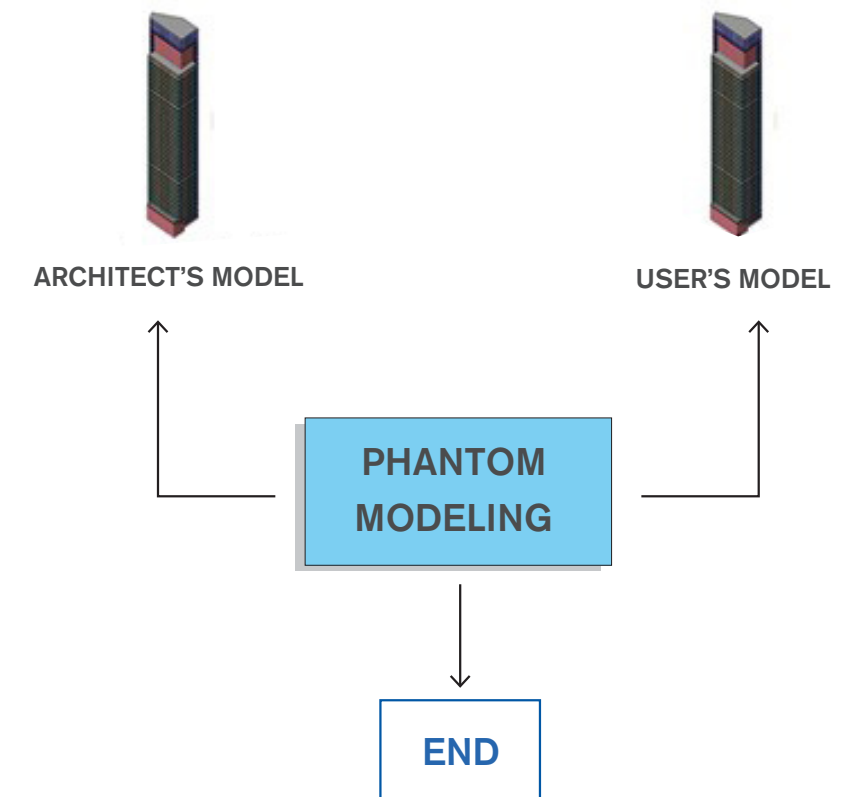


Figure 3. Conceptual interface relationship of phantom modeling.

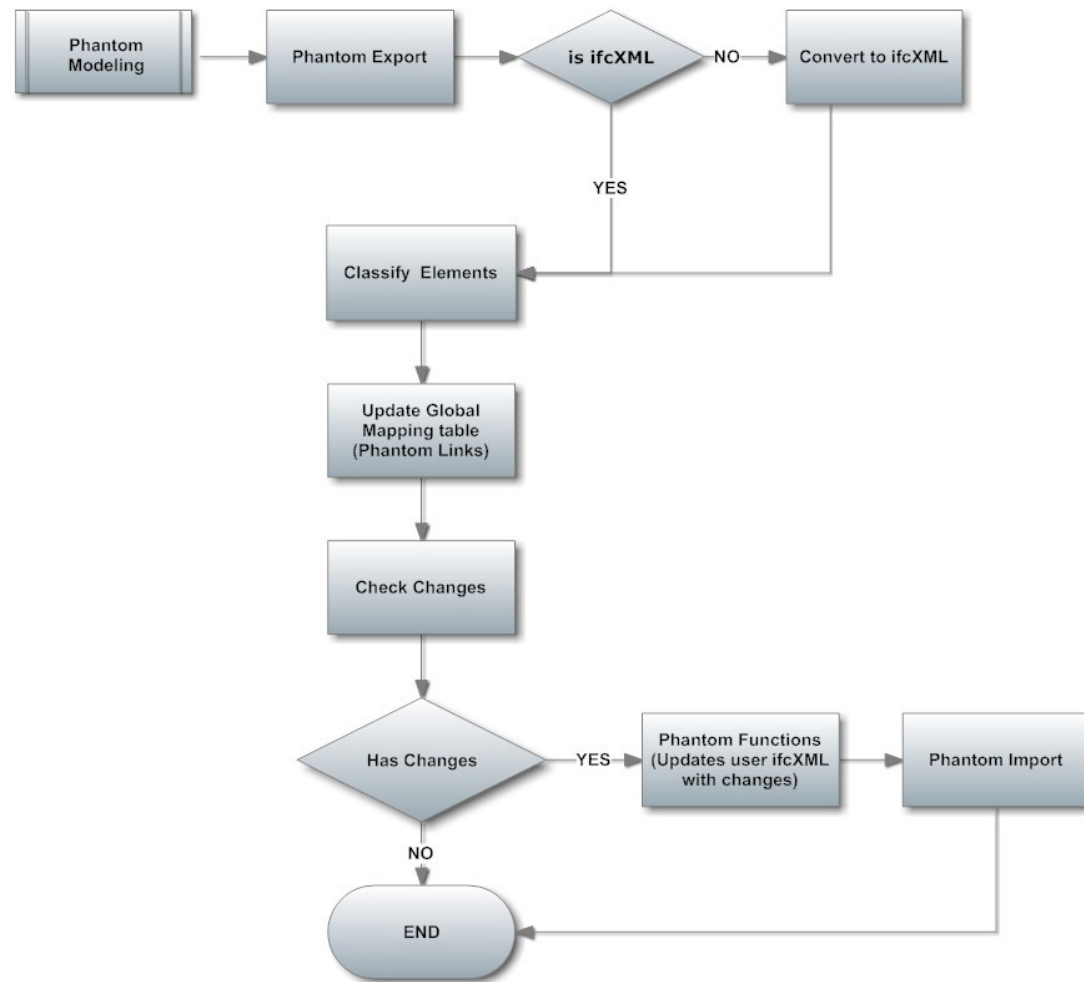


Figure 4. Internal process of Phantom Modeling.

INTERNAL PROCESS OF PHANTOM MODELING

1. *Phantom Export*: Export the models to ifc/ifcXML. The IFC model obtained from the architect's application is always transformed into native ifcXML, which will be validated against user defined XSD. Filtration can be done by selecting only the required building elements that you wish to synchronize, such as curtainwalls, interior walls, slabs, columns, beams, etc.

2. *Element Classification*: For more accurate model mapping, you can define

your own XML schema that contains all the rules, such as hierarchy and property definitions of various elements (in addition to their default "IFC type" definitions).

3. *Phantom Links*: The mapping table contains all the intelligent links, which includes version information and history of the links.

4. *Phantom Functions*: Phantom Manager allows you to list the eligible set of element classifications that are to be checked for changes, and to detect

IFC model version changes between the architect and user's model as a part of phantom model-based data exchange workflow.

5. *Phantom Import*: Once changes are made to the XML file, the new file is imported to the application, which will in turn synchronize the user's model. Since the incoming elements or modifications are converted into native format they become an active part of the architectural model, while retaining their properties (material, profile, etc) assigned to them before synchronization.

The transition from a paper-based exchange of design models to digital represents a substantial change in the design and construction industry. Digital models open the door for abundant automation possibilities, including large portions of the analyses done during the design phase.

CONCLUSION

The transition from a paper-based exchange of design models to digital represents a substantial change in the design and construction industry. Digital models open the door for abundant automation possibilities, including large portions of the analyses done during the design phase. With this comes potential critical consequences in regards to the speed and efficiency of future design processes that must be considered — ultimately, the quality of the resultant designs.

From a technical perspective, IFC and its use in the design and construction industry represents an interesting study for a number of reasons. The domain is challenging because of its breadth, and because of the size of its models. With the AEC industry being historically paper-based for information exchange and analysis, the opportunity for digital techniques to automate and streamline processes is significant. Working in such a highly collaborative environment makes interoperability a key issue, and the industry finds itself in a situation similar in many ways to the software engineering, where visualization-level interoperability has reached some level of maturity and semantic interoperability continues to develop.

The principal semantic interoperability challenges revolve around the quality and consistency of the models produced. Efforts are underway to provide for consistent modeling both through technical solutions and through the engagement of stakeholders to determine what constitutes good modeling practice. The success or failure of these efforts will go a long way towards determining the extent to which BIM succeeds in transforming the industry.

REFERENCES

- [1] Khemlani, Lachmi: "The IFC Building Model: A Look Under the Hood"; Building Science, Volume 28, National Institute of Building Science, Washington 2004.
- [2] www.iai-international.org
- [3] Thomas Liebich, IFC 2x Edition 3, Model Implementation Guide, Version 2.0, <http://www.iaitech.org/downloads/accompanyingdocuments/guidelines/IFCModelImplementationGuidV2-0b.pdf> (13 April 2010).
- [4] I. Hijazi, M. Ehlers, S. Zlatanova, T. Becker, L. Berlo.
- [5] Nour M (2009) Performance of different (BIM/IFC) exchange formats within private collaborative workspace for collaborative work, Special Issue Building Information Modeling Applications Challenges and Future Directions, Journal of Information Technology in Construction, Vol. 14, pg. 736-752, <http://www.itcon.org/2009/48>.
- [6] ifcXML Implementation Guide, Nick Nisbet, Thomas Leibich, Version 2.0, 2007.
- [7] buildingSMART, International home of open-BIM, <http://buildingsmart-tech.org/specifications/ifcxml-releases/ifcxml2x3-release/summary>.