Representational Limitations and Improvements in Building Information Modeling

Stefan Boeykens ¹, Herman Neuckermans ²
K.U.Leuven, Department of Architecture, Urbanism and
http://www.asro.kuleuven.be/caad
¹ stefan.boeykens@asro.kuleuven.be, ² herman.neuckermans@asro.kuleuven.be

Abstract. This paper discusses advantages and limitations of different representation types, illustrated with examples from current commercial Building Information Modeling applications. There is still a potential benefit in more thoroughly adapting additional representations to access and manage project data.

The paper presents arguments to adapt a hybrid approach, where multiple representations should form a series of interfaces to interact with a building model. Inspiration is derived from software applications not associated with Building Information Modeling.

Keywords: BIM; Representation; Design Software; Digital Building Model.

Introduction

In current Building Information Modeling (BIM) applications, the interaction with the building model occurs mostly through a combination of 2D and 3D graphical representations. However, a two-dimensional plan and a 3D model are both insufficient to fully represent a building. The building model, which is managed by the designer, does not have an unambiguous and complete representation. Both a 2D drawing and a 3D model can be derived from a building model, but they also form the two most important access points to that model. It is a goal for any BIM application that the different representations of the underlying model are at all times coherent. This is an important distinction with traditional CAD software, where drawings and models are directly created. The correlation between different views in such an approach is not maintained by the application but by the user.

Representational capabilities and limitations in commercial BIM applications

There are several commercial BIM applications available. This section focuses on three widely used applications. It presents some of their conceptual differences and focuses on current limitations in these applications.

AutoCAD Architecture (ACA)
This program, formerly known as Architectural Desktop (ADT), is a BIM program from Autodesk. It is an extension of the widely used AutoCAD software,
with architectural objects and an optimized workflow to divide a design into a series of DWG files. AEC Objects, which comprise all architectural entities that ACA adds to regular AutoCAD, are parametric. They have components or subparts, which have several properties and which can reference materials. Display Configurations define the characteristics of a certain representation, such as the drawing scale or the view type. The representation of AEC objects depends on the combination of the active Display Configuration, the current View Direction and the chosen Display Scale. At the same time, AEC Objects carry Display Styles. The default, global Display Style defines which components to show, with which graphical settings. Particular Display Styles can override the global style and adapt it to control how AEC Objects that share this style are displayed.

The Project Navigator defines the overall configuration of a building project, including information about floor level heights. Surprisingly, ACA lacks real floor levels, but uses externally referenced drawings (XRef). The biggest disadvantage is the lack of connections over different floor levels and the complexity to fully configure the Display System.

**Graphisoft ArchiCAD**

ArchiCAD is a dedicated BIM application, available for Windows and OSX, developed by Graphisoft, from Hungary. The main interface is a 2D drawing, which is shown floor by floor. Most modeling happens by placing building elements in this plan view, but at the same time, the full 3D building model is automatically created. Elements can be located over different stories, providing an adaptive representation for the 2D display. At any time, the user can open the 3D window and continue working in that environment. Changes all happen through the building elements, regardless of the active view type. Figure 1 displays the current ArchiCAD interface.

Library Elements generate geometry using the Geometric Description Language (GDL). Parameters allow a wide range of possible variations from a single script. Through the distinction between 2D and 3D, a plan representation could be symbolic while a full 3D object is generated for the 3D view. They can take the scale of the view into account, allowing for simplified representations on a rough scale, while displaying more detail on fine scales. This is comparable to the 'Intelligent Zoom', as discussed in (Neuckermans, 1992).

**Figure 1**

Screenshot of Graphisoft ArchiCAD, displaying different representations and project navigation.
Walls, floors and roofs can be composed of multiple material layers, which are shown in plans and sections. However, the 3D model still displays them as one layer. The 3D Document View is a recent addition from ArchiCAD 12, which allows a hybrid representation, with the detail from the 2D drawing, but displaying a projection of the 3D Geometry. There is no possibility to offset edges of individual layers inside a wall or floor composition. To correctly connect different elements, priorities are set, which define how intersecting layers make room for layers with a higher priority. Connections between multi-layered floors and walls are not solved automatically.

Solid Element Operators (SEO) provide parametric behavior between elements, using Boolean Operations (union, subtraction, intersection and two novel combined subtraction/extrusion operations). With SEO, a floor slab can create a real cut through a wall or a wall can be trimmed by a roof, allowing for cleaner drawings and correct volume measurements.

The representation model of ArchiCAD is rather complete, but connections or relations between elements are mostly non-existent. Scripted entities, in particular, live in a vacuum in ArchiCAD and have no influence on other objects.

ArchiCAD provides several hierarchic navigation windows, grouped in the ‘Navigator’ and its extended version, the ‘Organizer’, which assist the user to organize different output documents. There are other hierarchies available, notably when defining schedules or when adjusting properties of elements. There is no single building hierarchy however, apart from the division of the building model into stories.

**Autodesk Revit**

This is a dedicated BIM application from Autodesk. It is not based on AutoCAD and presents concepts which are more akin to Mechanical CAD software. Every single view is a representation of the building data, as shown in Figure 2. 2D Drawings, sections, elevations, plot layouts and 3D views are all directly connected to the building model. This is also true for schedules.

Relations between elements are enabled through the constraint system. Alignments, offsets and distributions are not only drawing aids, but can be embedded in the drawing, usually by enabling a lock on temporary dimensions. The model can actually be updated through editing these dimensions.
The Project View displays all available representations, containing stories, elevations, 3D views, sections, lists, families and output sheets. This preset and fixed set of top-levels divides the project according to the different representation types. Underneath each type, all available views are automatically added. However, just as with the previous examples, there is no way to show the full building structure as a hierarchy.

**Representing digital building models**

Representations are more than output documents derived from a building model. They are a translation of an abstract concept into a tangible graphical form. The design activity of an architect occurs mostly through representations.

In (Achten, 1997), the relation between knowledge and representations is elaborated. Graphic representations are at the same time encodings of the information that is represented. Encoded design information can be reconstructed from representations. A discrete set of graphic units is identified in a case study from different schematic drawings, to make a structured overview of generic representations. (Szuba et al, 2000) describe GraCAD (Graphbased tools for Conceptual Design in Civil Engineering). Starting from a graph model, the building program is translated into the listing of functional requirements. They are decomposed and used to generate a basic room layout, which is adjusted and checked against rules. The final result can then be translated into a CAD model.

The following section will elaborate different Representation types, explaining their usage, advantages and limitations.

**2D Plan Representation**

While a plan representation is, by definition, looking down from a slice through the building on eye-level, it is insufficient to simply cut through a 3D model. Some of the elements that would be invisible from the section are still represented. Examples include a beam above the current floor level or the contours of a floor opening, where a thin dashed line type is typically applied. Many other items are shown symbolically or the representation is enhanced with additional graphics.

The 2D drawing does not contain sufficient information to fully describe a 3D building. Certain corners or details are simply not visible in floor plans or sections, making it difficult and error-prone to derive 3D information. It is also difficult to represent freeform building elements in a 2D drawing. Simple curves can be accurately described in 2D provided the curve is parallel to the drawing plane, but double curved surfaces can not be represented correctly at all in a flat drawing.

However, different academic research projects prove that there is still interest in developing approaches to combine manual drafting and sketching in a digital context. EsQUIsE (Juchmes and Leclerq, 2004) uses video capturing to interpret sketches. In the work of (Gül and Maher, 2006; Maher et al, 2006), a 2D sketch environment is augmented by a 3D world. Apparently, the exploration of abstract design concepts motivates the use of 2D sketching techniques, whereas 3D is used for elaboration of these concepts. These studies show the value of both 2D and 3D design methods.

**3D Model Representation**

A building is represented with 3D geometry, such as prisms, boxes, extrusions and Boolean operations between geometry. While a building is an assembly of physical objects, the 3D Model of that building is a simplification. Depending on its purpose, the focus shifts from representing textured or shaded surfaces for visualization, to volumes of common building elements for quantity calculations or simulations.

When modeled accurately, it is fairly easy to extract volumes and other quantities. Creating a section through the model is also straightforward, using clipping planes. Figure 3 displays a 3D model that has been used to generate a section. The part of the 3D model that is displayed in wire-frame is not
shown in 2D.

However, slicing or clipping alone will not generate the complete graphical output required for architectural drafting. In fact, the creation of a 2D section or elevation has always been the result of graphical expertise, to display accurate information, but in a legible way.

The main limitation is the need to model everything that should be visible or extracted from the model. The model has a limited level of detail, which is usually fixed. Elaborating a more detailed building phase might require the remodeling of the complete model. The construction detail is usually not modeled, but represented with additional detail drawings in 2D.

Hierarchic Representation
A Hierarchy represents a certain order and certain dependencies. Hierarchies can be found at multiple levels, from the organization structure of most countries and economies over classification systems to computer systems.

In the context of building models, where the model serves as a spatial model, containing building blocks, floor levels and rooms, it seems appropriate to introduce a building hierarchy. Most BIM applications provide a combination of hierarchies for view types and drawing output, but the hierarchical organization of the building model itself is usually lacking, or limited to a simple hierarchic structure of floor levels and sometimes zones to subdivide larger buildings.

The representation of the hierarchy in a building model is better supported in applications built around the Industry Foundation Classes (IFC), managed by the International Association for Interoperability (www.iai-international.org). This is a neutral file format for the exchange of Building Information. Standalone IFC software, such as the IFC Engine Viewer (www.ifcbrowser.com) or the Solibri Model Viewer (www.solibri.com/solibri-model-viewer.html), presents both a graphical view of the 3D model and a hierarchic view of the IFC structure, as shown in Figure 4.

Figure 3
A section through a 3D Model to generate a 2D Plan (edited screenshot)
While IFC is currently supported in all common BIM applications, a program such as ArchiCAD presents the IFC structure only from within the IFC-support add-on, making it completely different from the main software. The IFC structure uses the term ‘containment’, which defines ‘part-of’ relationships. E.g. furniture is placed inside a space, which is located on a particular floor level or as part of a building block, which in its turn is a part of the building project. Figure 5 displays a generic tree-structure to clarify the structure of a particular design.

It is important to understand that it is not possible to define a single hierarchy that suites every design project. Some spaces are part of multiple floor levels and some designs are not clearly divided into floor levels at all. It is mostly the responsibility of the architect to define a suitable hierarchic structure, while the BIM software should provide flexibility supporting this structure.

**Schematic Representation**

One of the visual schemes of particular interest for an architect is a diagram. Instead of making a graphical representation of elements according to their size, each element is represented by a symbol. Relations between elements are represented as lines or arrows. A small set of shapes, line types and arrow styles allows for a semantically rich representation of a potentially complex set of data elements.

It makes sense to provide such a representation in an architectural design application. A schematic

![Figure 4](image1)

**Figure 4**
Solibri Model Viewer, displaying the building structure from an IFC file

![Figure 5](image2)

**Figure 5**
Example of the display of a hierarchic building structure

![Figure 6](image3)

**Figure 6**
Example of relations between spaces in a Schematic Representation of a design project
Figure 7
Different representations can be used to access a digital building model

A diagram can depict relations between spaces, combined with circulation and visibility. It would even be interesting to define the whole building program schematically, by defining a set of requirements and a proposal of a scheme. The layout of the diagram allows querying the design with regard to adjacency, circulation schemes, visibility and connectivity, as illustrated in Figure 6.

Even though this visual representation is a valid, albeit partial representation of a design, it is not used in architectural CAD-software. In contrast, a schematic or diagram type of interface can be found in applications as varied as Microsoft Visio, Demicron Wirefusion and applications such as Autodesk 3ds max or SideFX Houdini, which all have a schematic representation as one of their view types, where relations between elements can be managed. Different multimedia authoring applications, such as PureData or Apple Quartz Composer provide a similar interface, depicting streams of data, such as audio, video and controller data to be managed by the user.

**Textual or Tabular Representations**

Architects often need to summarize information from a building project in tabular format. Examples include quantity takeoffs and the bill-of-materials. These listings, which are often considered as output, are in most cases elaborated separately from the drawings. In a BIM context, most applications nowadays allow custom listings to be generated from the model. Some building data can even be updated from the tabular ‘output’, promoting listings as an additional interface to the building model.

While it sounds obvious that editing a number in a list might also modify the element from which this number was calculated, it is not always a trivial task. Imagine the adjustment of the composition of a wall, by increasing the thickness of the isolation layer. This will be reflected in all connections involving walls that reference this composition. Editing one entry in a list might also trigger several other places in the global list to be updated. We might look at this task as a detail-view in a database query. The list can be interpreted as a form, resulting from a query. Adjusting fields in the form will update the data and a subsequent call of the same query will show the updated listing.

**Choosing a hybrid representation scheme**

Instead of being limited to typical 2D and 3D Representations, BIM software should present the user with a choice of representations, each displaying a part of the project data in an optimized form, as illustrated in Figure 7.

The digital model can not be a complete faithful duplication of the physical building. This motivates the dashed line in Figure 7, which separates the building from its digital model. The important goal is storing enough information to be able to generate other representations and to make informed queries on the model.

**Conclusion**

The different applications which have been discussed in this article each have their advantages and disadvantages. In the context of the development of BIM software, suggestions are made, to improve the capabilities of the software, beyond their current approach, oriented to construction documentation.

BIM software should allow multiple simultaneous representations, which are all linked to the same building model. It is important to also support hierarchic and schematic representations, next to the
obvious 2D and 3D graphic representations. This paper hopes to motivate their elaboration and provide better insights into BIM application development, to become more supportive for the architectural design process.

References


