CAD Smart Objects:
Potentials and Limitations

Magdy Ibrahim, Robert Krawczyk, George Schipporeit

College of Architecture, Illinois Institute of Technology
http://www.iit.edu/~Ibramag, http://www.iit.edu/~krawczyk

Abstract: For many years, CAD software depended on entity objects that were manipulated and interpreted by the user as meaningful symbols. These entities only represented the geometrical aspect of the design, but never had knowledge of what they are, or how to behave. With the new CAD systems, this concept has changed into the smart CAD objects. The smart objects will automatically provide all the data related to it: geometry, materials, specifications, price, as well as manufacturers and theoretically any related data.

Creating new objects is not an easy straightforward job, and requires more programming skills than previously needed. Taking into consideration the relative difficulties in learning to modify and create new CAD objects, this might lead to a new branch of learning, as the architecture students might not only need to learn how to use the CAD packages but also how to program it in a way that makes them capable of doing what they want rather than doing what the package allow them to do.

Keywords: CAD; objects; object oriented programming.

Building Information Modeling: The New CAD Concept

BIM stands for Building Information Model. In fact, the idea of BIM-based CAD is not new, it has always been foreseen as the ideal way to represent the building digitally, but it has never been mainstream for commercial products until recently; mainly due to the increased capacity of the personal computer. Graphisoft ArchiCAD Virtual Building concept may have been the first commercially available package that utilized the building model; now more and more CAD software is being built around this new concept. Object oriented programming is not new also, all applications written in C++ are object oriented, but it took the CAD industry some time to apply this software concept to the building elements themselves.

The building industry has traditionally communicated building construction information through drawings with notes, schedules and specifications. CAD technology automated the production of that process. The result of earlier manual drafting and CAD systems were identical, it was to create graphic abstractions of the intended building design. These systems were intended for generating two-dimensional printed/plotted drawings, and were capable of handling and managing information about a building only on a limited basis. This BIM-based generation of CAD systems, designed with current technology, is required to fully realize the benefits of object oriented CAD. This next generation of information-centric software provides building information modeling instead of building graphic modeling.

BIM operates on digital databases. By storing and managing building information as databases, these systems can capture, manage, and present data in ways that are appropriate and customary for a particular designer, contractor or vendor. Such applications start with capturing and managing information about the building, and then present that information back as conventional drawings view or in any other appropriate way such as tables or perspective views, and make it available for use and reuse at every phase in the project. (AutoDesk 2002)

With the arrival of the BIM based CAD, a new concept of objects has also arrived. These objects are not only programming objects, but they have specific meaning to the architect. They have an equivalent physical meaning to real world objects, and provide an abstract computer representation of the physical world that is convenient for architects (Ruppel, Meissner, and Bernd M 1993). A wall as an object in the CAD system represents an actual wall in the physical world, and a door as an object represents a real door.

With objects, all the standard object-oriented programming concepts apply. Objects have properties, methods, and events. The advantage of the object model is that it allows for the extension of the properties or attributes, not to be confused with the block attributes as found in AutoCAD.

Entities vs. Objects

An object from a computer science point of view is an independent procedure that contains both the instructions and data to perform some task, and the programming code necessary to handle various messages that it may receive (Morris 1999).

AutoCAD is an example of a C++ written object-oriented program which used a general concept of objects to create the “drafting elements” or “drawing primitives” such as lines and arcs. While AutoCAD itself is an object-oriented program, the objects it provided were only graphical objects or “entities”. Although such objects had all the concepts of programming objects, they were used mainly to draw a representation of well-understood drawings of highly symbolic information about the building. The architect has to interpret the meanings of what has been drawn, in the exact same way as with physical drawings. Using the same drawing legacy, a replica of what could have been drawn by hand was created using the computer as a drafting system. Even the creation of symbols library, blocks in AutoCAD, is very dependant on the previous knowledge of the symbols used by the profession and to some extent equals the use of drafting templates.

These drawing entities only include geometrical aspects of the real objects that they represent, and never had knowledge of what they are, or how to behave or interact with each other. With the exception of Blocks with Attributes and their equivalent in other CAD packages, which to some extent had included more information about themselves, all entities were just a collection of basic drawing primitives.

With the BIM generation of CAD systems, embedded information can describe the geometry, as well as, materials, specifications, code requirements, assembly procedure, price, manufacturer, vendor and any other related data associated with how the object is actually used. A door as a smart object understands its relationship to a wall and reacts accordingly should be a great help to the designer. The potential of using CAD smart objects is very appealing in the production phase of a project. Although the architect is not obliged to give full information about the object he is using, that object has blank attributes waiting for that input. As a result, in the schematic design phase an object could be represented symbolically to the architect, and as the project advances through design phases, these semi-defined objects would become better defined as more decisions are made about the building.

In the production phase, objects are in full throttle, generating bill of materials is a matter of a button click, and consequently the cost estimation is easily achievable. Any changes done to the building database will simultaneously be reflected in the entire set of documents the architect is responsible for: plans, elevations, sections, schedules, and bill of materials. All types of engineering analysis could also be performed.

How will this Change the Profession?

The smart CAD object concept, as remarkable as it is, might have a set of negative implications:

1. The Master Apprentice Relationship and CAD

A model of the smart objects was presented (Eggink, Gross, and Do 2001) suggested the object would change its parameters depending on its boundary conditions; a beam-as an object-should set its depth according to the span it covers, and change accordingly as the span changes. Although such intelligent behavior has not been implemented yet in current CAD programs, it is this type of development that will be expected in the future. Another example, a door object should set its fire rating automatically according to the room where it belongs depending on its knowledge of this room's function and the code requirements for this function. Theoretically, this is an achievable task, and should enhance quality control in all phase of the project.

Well developed CAD objects enable embedded information to be the primary reference for an architect, placing less emphasis on the master apprentice model that has developed in architectural practice. The young architect will not need much technical support from his seniors as previously required as long as the computer provides this feedback for him. Will this lead to a new kind of designer, who might not require much help and will complete a task faster and with relatively fewer mistakes? But how this will affect developing problem solving skills, and the master apprentice relationship in the workplace, promotion within a firm as well as the professional education of an architect?

2. Customization

For a design firm, utilizing this kind of technology means savings in time and resources needed to coordinate changes. Even with advanced current CAD methods, like external references and writing routines to automate drawing production, this would have not been achievable with the ease and accuracy a BIM-based CAD could deliver. This can lead to a problem where the customization of CAD packages may not be possible. Technical support teams in most firms now can easily automate repetitive tasks, and develop even more intuitive tools to help in their production. With the new breed of CAD that depends on smart objects, the customization capabilities of the older and simpler entity based systems might be in question. It can be foreseen that the same need for customization will take place no matter how sophisticated the software becomes, but the inherent complexity in the new systems may make it more difficult or not possible.

3. Evolution of Objects

How are objects represented during the different phases of the design process? From the schematic design to the design development, they need to have empty properties that might be added while the design is being developed. Can these properties/attributes be added dynamically as needed? It is possible to make the objects dynamic enough to expand its knowledge as needed, this will make them like containers which will keep filling up with information during the different design phases. This should not be limited to a previously determined set of parameters, but to the possibility of adding more parameters as needed. The architect should be able to add more specifications to the object as parameters not just as textual description. There is no indication that current developers have addressed this issue.

4. Innovation and Creativity

Should smart objects allow for nonstandard uses? The smart object concept may not allow the architect to assemble non-compatible components together. How this might affect innovation? Pre-defined rectangular shape of a door opening of an object is normal, but what about the other door openings shapes? The current software does not allow architects to create their own non-orthodox type of objects, unless supported by the vendor. When Frank Gehry selected a suitable CAD package for his practice with forms that do not follow traditional shapes, he used CATIA, a non-architectural high-end CAD system that is meant for product development. There is no such versatile tool specially tailored for architects that gives them the same freedom and functionality while utilizing the objects concept.

What will the support group within a design firm need to know about the software used in order to be able to change its parameter to support more untraditional concepts?

5. Dependability on the Software Industry

If firms are not able to add new objects easily, will they have to wait until the next version of the package to come to market? The obvious answer is yes. This is what is happening now, having to wait for a curtain-wall tool in AutoDesk ADT, and the race between the other companies to provide the same feature to its users to keep them competitive. This kind of competition is healthy as the end user is the most beneficial, but in the same time, it is the architect who has to wait for a tool to be implemented. Consequently, the architect has to pay more money to get that new feature through upgrading. There might be a point in time in the future where these systems will become comprehensive enough for most needs, but there will always be some special needs that are not foreseen.

6. Architectural Education

The architectural student might need to be educated differently to cope with this kind of change in the tools, not only being able to use it but understanding the potentials and limitations of the tools as well, and above all having a good understanding of the programming theories behind these computerized tools. Those students will constitute the users for such systems, from both ends: architects and IT support. This might mean more programming classes for the architecture student who might have the potential of becoming one of the support team. Not only will the future designer have to be good in using CAD software but also in programming and software development.

References:

AutoDesk. (White paper). (2002). Building Information Modeling. Autodesk Building Industry Solutions. San Rafael, CA: Autodesk, Inc.

Beucke, K., & Rang lack, D. (June 1993). Computing with objects: What does industry hope to gain from it. In Louis F. Cohn, Computing in civil and building engineering: Proceedings of the Fifth International Conference (V-ICCCBE) Anaheim, California: The American Society of Civil Engineers.

Eggink, D., Gross, M. D., & Do, E. (29-31 August 2001). Smart Objects: Constraints and Behaviors in a 3D Design Environment. In Hannu Penttilä, Architectural Information Management: 19th eCAADe Conference Proceedings Helsinki (Finland): Helsinki University of Technology (HUT). (Pp. 460-465)

Hartmann, D., Fischer, A., & Holéwik, P. (June 1993). Object Oriented Modeling of Structural Systems. In Louis F. Cohn, Computing in civil and building engineering: Proceedings of the Fifth International Conference (V-ICCCBE) Anaheim, California: The American Society of Civil Engineers. (Pp.78-85)

Marty Rozmanith, Product Management, Autodesk Revit. (White paper). The Parametric Building Modeler: Answers to Technical Questions.

Morris, S. (1999). Object-Oriented Programming for Windows 95 and NT. Digital Press.

Revit. (White Paper). The AEC Technology Platform Question.

Rudolph, D. (December 1999). Mastering AutoCAD 2000 objects. Alameda, Calif: Sybex.