3. EXPERIMENTS IN ARCHITECTURAL DESIGN DEVELOPMENT USING CAD

Charles M. Eastman and Jurg Lang

Graduate School of Architecture and Urban Planning
University of California,
Los Angeles, California, U.S.A.

The need to explore development techniques in computer-based design is reviewed. Some premises are given for design development using computers, including integrating multiple representations, the use of object-based modeling and the importance of visual analysis and 3-D modeling. We then present techniques used in a UCLA design studio that explored methods of computer-based design development based on these premises. The two main methods used were hierarchical object structures and multi-representational coordination. They were applied using conventional CAD systems. Some lessons learned from this class are reviewed.

Introduction

During their professional education, future architects learn in their studio classes alternative processes of design. They learn from their teachers the use of different representations, such as concept sketches, bubble diagrams, circulation diagrams, massing diagrams as a means to analyze certain compositional issues, or the use of wall sections as means to study fabrication and detailing.

In the sequence of assignments, students also learn by example various sequences of design development using these representations. For example, a studio may require circulation and massing studies in the early weeks, followed by a site study, followed by a schematic set of plans and a model. Of course, different faculty utilize different sequences, possibly for different types of projects, but each series of assignments indicates to the student alternative development sequences for the evolving design. After experiencing these imposed processes for a few iterations, they become "natural" and are incorporated into the way architects think about design. In this manner, early teachings become the natural and unself-conscious mode of working in later practice.

This type of architectural education has existed since the beginning of architectural schools. The processes have been refined and have evolved in a natural way, resulting in an effective, though largely uncodified, set of practices.

Notice that most of these practices are intricately woven around the use of paper-based, iconic representations. Plans, sections, isometrics, elevations are intrinsic to these processes and the mode of thinking they build upon. The physical model has been the only important alternative medium of design consideration. Notice also that engineers have strongly moved into algebraic, symbolic representations, quite different from those used by architects. That is to note, other design professions did not follow the architectural path.

Over the last ten years, computers have reached a stage of development and a price where they are thought to serve as alternatives to the more traditional paper representations. It was
naively thought by many in the 1970s that computers would quickly replace drawing boards and that new computer-based design processes would quickly evolve. That has not happened. The computer instead has made significant inroads into architecture by providing one-for-one replacements for earlier representations, particularly orthographic drawings, renderings, energy and other forms of analysis, and recently physical models. If one learns computer-aided design (CAD) today, or buys a CAD product, the functionalities provided are those that support a substitution for some existing representation. They are also the main focus of current computer marketing. This kind of one-for-one replacement allows computers to be integrated into design without re-work of the entire process. The informally learned process of design can continue and architects can work in their traditional modes with relatively small adjustments.

The computer however has different operations and different representational strengths from paper or physical models. In the longer run, one can expect quite different processes of design development to evolve. Thus there is tremendous need to actively explore larger re-organizations of this process. Such re-organizations are usually not practical to undertake in architectural practice, when under the fire of deadlines and guaranteed results. Thus the schools of architecture have a major contribution to make in re-thinking the process of design in light of computer based design tools.

Computers are still a scarce commodity in schools of architecture and many students pass through schools today without gaining familiarity with their use. Most schools hope to impart some level of competency in their use through a short sequence, possibly culminating with a single studio emphasizing computational techniques. At the same time, the student still takes six or more studios utilizing paper and cardboard media. The integration of computers in architectural students design processes has barely begun.

**Theoretical Premises**

Traditional design involves working in multiple representations. These are usually different forms of drawings: plans, sections, elevations, drawn as schematics, design development or construction drawings. Other representations of an architectural design include, physical models, specifications and the data used in various analytical calculations.

The typical process of design involves using one or more of these representations to develop a design concept. After this has been accomplished, the design information is translated into other representations to gain insight with regard to its physical form, functional performance, cost or other issues of interest. Design proceeds with the eventual definition of an array of representations, many generated by specialists, such as structural, mechanical and lighting engineers. Each new representation informs the existing ones. Eventually, all the representations consistently define a single building or artifact that satisfies the various intentions for the project.

From this description, it is clear that an inherent aspect of design activity is the translation between representations. We are most aware of this when we make a physical model from plans and elevations, or when we take data from drawings for input into some form of analysis. Most of the time, these sorts of translations are expensive and time consuming and are therefore done only occasionally. Many representations that could inform the architect on some behavior of the design are not used during design development because of the cost of data translation between representations. These include both computer-based
representations, such as energy or lighting analysis, and others that are manual, such as maintaining a physical model of the current design. Most representations are used late in design, as a check that it does not perform badly rather than to inform how the design can be improved.

**Visual Analysis**

An important architectonic example of this issue is visual analysis. What we call visual analysis is the development of a special representation of the design in order to better evaluate its appearance or experiential impact. Studio faculty commonly require physical models and renderings as representations of design work, so students may gain experience in evaluating design in these representations and integrating them in their development processes. Some faculty teach good methods of model refinement and modification. Yet, physical models and renderings are time consuming to generate, and like all others, are typically used toward the end of schematic design.

Computer based visual analysis should be a commonly used capability, given the importance of visual considerations to the profession. Computer modeling of buildings in three dimensions and the generation of high quality images is a well developed capability. Many presentations at CAAD conferences show the stunning results that can be extracted from modern computer workstations running advanced visualization software (Greenberg, 1989). When examined, however, most modeling has been done as a rendering, at the end of design, for presentation uses, in a manner replacing a physical model. Exploration of design alternatives have still been done manually, with little or no computer aid.

While there is growing exploration of 3-D modeling and visualization in schools, there has been little application in practice. This is because 3-D modeling typically involves a separate representation from a set of CAD drawing data, and separate development. School studios can encourage design using 3-D modeling exclusively, without considering the need to generate traditional drawings. In the profession, such strategies are not practical. Firms only use 3-D modeling if they can charge extra for defining the model, similar to construction of a physical model (Teicholz, 1988). It has not yet become integrated into normal design development practices that produce drawings as the main product. Some computer advocates propose that eventually drawings will disappear and a 3-D computer model will serve as construction information. That will not happen soon.

In detail, there are three issues being addressed here. First is facilitating the use of multiple representations in design, by reducing the cost of translation. The computer can make a major contribution to this issue, but it requires new processes of design development. Second, the integration of 3-D modeling techniques to design development is central if other geometric data is to be generated for other representations, such as lighting and visualization, energy, acoustics or structures. Only 3-D modeling supports the generation and checking of geometry in a complete manner (Eastman, 1976). Visual analysis, then, is the lead application to integrate 3-D modeling with drawing production, and opens the door to integrating other applications. Computer based 3-D models and visualization should allow visual analysis to be used throughout the design process, leading to better designs.
Object Based Modeling

There has been continuous interest in what we call "object based modeling" of buildings. Object based modeling involves representing a building in terms of the objects that it is composed of. An object is both a unit of composition and a carrier of different information used in multiple representations.

An object may be any unit of composition, ranging in scale from a complete building shell, to a wall, to a window pane. Multiple representations are defined for each object. For example any such object may have a plan, several elevations, and a 3-D model, as well as associated property descriptions, each of which is also a separate representation. An example is shown in Figure 1.

Objects are organized hierarchically. Highly aggregated objects, such as an apartment unit has as parts floors, ceilings, walls, stairs and so forth. See Figure 6. Each of these objects also have lower level objects. The hierarchical relations may be defined and managed explicitly by the CAD environment. Alternatively they may be managed by the user, who keeps track of what objects are composed into other objects at different levels of abstraction. The window pane is known to be a part embedded in some wall; a wall is part of some building unit and so forth.

A second aspect of object based modeling is the logical structure it suggests as a framework for design development. A design, defined as a hierarchy of objects, can be modified in several ways, each having implications with regard to maintaining the hierarchical structure. A change may be made to any object at any level of detail, with propagation upwards in the hierarchy to more aggregated objects and also downwards in the structure to the detail objects the change effects. See Figure 2.
FIGURE 2: A graphical representation of the different types of actions that are embedded in the object model approach to design.

Three types of development operations are naturally defined:

1. A change to the arrangement of objects can be made at a top or middle level in the hierarchy, resulting in changes to both the higher level aggregated objects of which it is a part and also the lower level components. For example, a different arrangement and shape of walls is defined for an apartment unit, resulting in a change in the area of the total unit, its relations to site, the total building shape and other aggregated properties. It also requires changes to the components making up the wall, i.e., downward, to realize at the detailed level the new shape. These may be achieved automatically, if they simply involve spatial transforms of existing component objects. If shapes or properties are modified, then the designer may have to determine how these changes will be realized at more detailed design levels.

2. A second kind of change is made by replacing one object for another; this change eliminates all the objects below the previous one and replaces them with new ones, if they have been defined. It also propagates upward, resulting in new aggregated properties and/or geometry.

3. Also, an object at the bottom level can be further decomposed and refined further. For example, a wall may be defined in terms of its internal materials, surfaces and molding.

Object based design has a powerful logical structure that can support a variety of development processes. While it is based on a well-defined set of parts, the designer can define and re-define the parts as the design proceeds.
There have been a few examples of object-based CAD systems, both in research: GLIDE (Eastman and Henrion, 1977) and Arch:MODEL (Borkin et al, 1976), and also commercial systems: GDS, STAR and SONATA. The commercial systems embed most of the capabilities described in an explicit fashion. Object based design provides at least one important design development paradigm for architecture.

Object based modeling can be undertaken, using the traditional CAD structures of symbols¹ and layers (Eastman, 1991). A symbol is used to describe and store an object. An object may be abstract and conceptual, such as a building mass or activity space; or it may be detailed and correspond to a component used to construct the building, such as a door or window.

Using symbols to implement object based modeling requires a different type of layer management scheme from those typically recommended (such as by the American Institute of Architects). It is important to allocate a separate set of layers for use by different classes of representations early in the process and fix them: for example, plan, section, 3-D model and elevation. All symbols will use this structure (as they do with normal layer schemes now). These sets of layers need to be subdivided further to support: various scales of presentation and level of detail, and different design disciplines, similar to good use today. An example layer management structure is shown in Figure 3. These allocations are used to manage information in all symbols.

Layers are used to manage graphic information. Another allocation is needed for attributes. In most CAD systems one can define multiple attribute sets for an object, allowing each attribute set to be used in the same way as layers, but for data associated with some form of analysis.

This approach is based on the recognition that, using todays drawing conventions, floorplans and elevations are different from an horizontal section or front projection of a building. Sections of a 3-D solid representation require extensive editing to be used for floorplans: line weights are different, dotted lines show ceiling conditions on the plan, stairs and cabinets are abstracted. These differences are reviewed in detail in (Eastman, 1987).

---

¹ also called figures and blocks in some CAD systems.
**Intention**

Like most schools, U.C.L.A. offers a sequence of classes applying computers to design. After two quarters of background classes, students take a studio devoted to design development, emphasizing computer-based representations. These studios present to the student some known methods of design development using computers and allow the students to explore others on their own.²

This paper presents the results of one such exploration, developed in the context of a design studio taught in the Spring of 1990. Students were at different levels in their architectural education; some were in the second year of a first professional degree program and others were advanced students taking a second degree specializing in CAD.

The students enrolled in the studio had already taken previous courses in computer use. They had all surveyed the effective use of the standard computer-based representations, especially 2-D orthographic drawings and 3-D modeling for visualization. Some knowledge of managing multiple representations had also previously been presented.

In the spirit of traditional studios, the faculty (the two authors of this paper) presented a design problem for the quarter and then defined a set of required submissions that were based on pre-developed concepts of computer based design development. That is, the sequence of submissions were organized to suggest a development process uniquely useful in the use of CAD for design. Otherwise, the course was approached as a traditional design studio, focusing on architectonic issues.

An important aspect of our studio methodology was to explore how to use 3-D modeling and visualization as an integral design tool, while still producing drawings. At the system level, visual analysis was supported in two different ways: (i) in simple format using the 3-D capabilities of the CAD system; and (ii) by extracting data and sending it to a special visualization computer, allowing realtime walk-throughs of spaces. The first method was used for generating the layout, the second allowed high quality, quickly generated visual reviews.

![FIGURE 4: Some of the different plans developed in the studio that reflect different strategies for gaining amenity for each of the residential units.](image)

---

² U.C.L.A. is currently revising the CAD part of its program, with a required sequence for all students and broadened integration, both in design and technology classes.
The conceptual goal is to allow parallel development of the design in multiple representations, so that design actions always set up the data in the various representations to be ready for use in each of them. Visual analysis was considered as a test case for other forms of analysis, such as energy, lighting or structure. If integration could be achieved for visual analysis, the same techniques should transfer to these other forms.

The CAD system used was Computervision CADDS4X running on SUN-3 workstations. 3-D visualization while designing was supported by the surfaces and solids packages of Computervision. Visualization analysis was supported by the Personal Visualizer software developed by Wavefront Technologies running on a Personal IRIS workstation by Silicon Graphics. Translation from Computervision was accomplished by using a CV IGES translator out and Wavefront IGES in.

Examples / The Project

The studio project dealt with medium density housing in Los Angeles. The site is one of the largest remaining tracts of open space within the city of Los Angeles, the old Howard Hughes airport site at Playa del Rey. The site, consisting of about 700 acres, bordered on the south by the Loyola bluffs, on the west by Santa Monica Bay and on the north by Marina del Rey, is currently undergoing development. The developers are attempting to make it a model for new development within Los Angeles. As such, it is of higher density than most housing. It is to include residences for a wide range of income and family types. It includes neighborhood shopping, schools and recreation, at a density that encourages access without using automobiles. It is foreseen that some blocks will have convenience stores and professional offices along one street.

![A floorplan of one of the units and the set of walls that bound it.](image)

There were strong design guidelines regarding street widths, sitelines and setbacks. The goals embedded in the guidelines included a strong street facade with a small but significant buffer from the street, wide walking avenues and minimal on-street parking. Most parking
was underground. The program implied walk-up units in buildings of two to four stories in height.

Each student took one block as a design and development area, which consisted of between 2.5 and 3 acres. The program density ranged from 35 units to 80 units per acre, with units ranging in size from 900 square foot studios to 1500 square foot two-bedroom condominiums. All student projects involved multiple buildings on the site with one or more interior courtyards.

**Multiple Representations**

In the previous class, students learned to work in object-based modeling, using traditional CAD systems. In that class, the focus was on de-mountable buildings which had a natural affinity with modular construction. That was not the case in this project.

The bottom level elements of composition used by students in this project were walls and other panels, which could be composed into living units, or else the housing units as a whole. At the unit level, explorations were undertaken regarding how single units could be composed in various ways to gain additional amenities for the units, as shown in Figure 4. Pinwheel layouts were tried, as a means to gain private outdoor space, as well as split-level plans, where each unit consisted of two floors, not stacked but staggered diagonally about a central entrance. In this form of layout, the expanse is greater than a stacked plan and cross-ventilation is also provided.

After the units were defined, they were made into symbols. The plan was defined on one layer, (shown in red) and the walls were defined in 3-D on other layers, as shown in Figure 5. In some student's projects, the interior walls were separated from the outside walls by assigning them to different layers, so they could be turned off when outside images were generated. The result was that each object corresponded to a symbol in the design. Changing the symbol or its location changed its location in all the representations, here, 3-D and plan.

The units were also composed into a building. See Figures 7 and 8. Street facades could be evaluated from many angles. Students found that if they took certain elements out of the unit and located them at the building level in the hierarchy, this allowed them to be varied for different units, and respond to the composition of the overall facade in rich and interesting ways.

With the exterior walls defined for each unit, and in some cases, stair and balcony railings and exposed framing, the amount of visual information in the design became quite large. But because each unit or detail was a symbol, only a few instances were stored in full in the project database³. Regenerating the design from the database, however, could be tedious.

---

³ Computervision requires that shaded images use Extended N-Figures, a type of symbol that copies all of the components. Thus symbols do not realize much of a saving when using CV alone. The IGES translator, however, allowed N-Figures to be translated and viewed on the Personal Visualizer, which did allow the savings.
FIGURE 6: The composition of a unit into a building. The single unit floor and walls had associated facade walls and alternative window units. Atriums were added and the unit composed into a building, which was then composed into the complex (project by Ken Aluko).
FIGURE 7: The unit developed in Figure 6 was composed into a building complex, with varied massing and details (project by Ken Aluko).

Because the composition was composed of symbols and all their layers, it was simply a matter of turning on the appropriate layers to generate the building floorplan, or alternatively, another set of layers for the 3-D model. Alternatively, both could be shown together. See Figure 5. Using this process as described, party walls were duplicated. This could have been eliminated by making the potential party walls separate objects and including them or not, as required, within the overall building configuration. In an office environment, additional representations would have been added to the symbols, for elevations or sections.

By grouping the multiple representations of an object within the design on the different layers of a symbol, the coherence of the object data is maintained. It does not eliminate the need to propagate a change made in one representation to the others, but it makes this job easier because all the representations associated with the object are together. And of course, a locational change to an object operates on all views.

The effect of this approach was that students were able to develop their design in a way that produced both drawings and 3-D images throughout the process. Some students used the visualization capabilities of Computervision for quick reviews of their composition. Each view usually took several minutes to compute. Others found that they could translate their data and bring it up on the SGI Personal Visualizer in about the same time. But once in the Personal Visualizer, its Quickshade option allowed any view to be generated in just a few seconds. They could walk through their design (in wire frame mode) select a view and have it
shaded in a couple of seconds. One student commented that though he had used CAD previously, this was the first time that he gained significant insights into the design faster than by using paper and pencil.

**FIGURE 8:** Another design developed for Playa del Rey. Here the student refined the external character of the building over time, and developed a roof garden scheme where each unit had its own garden, on the ground or on the roof of an adjacent unit.

Another effect was the elimination of most of the time traditionally needed for presentation. Most students had practiced generating good rendering quality images as they carried out their visualization studies. As a result, most of the students generated their presentation images in just a few hours, allowing much longer time to refine the design.

**Hierarchical Development Strategies**

Space savings was not the primary benefit of making the living units and all other objects into symbols. If all objects had been copied rather than instanced, the common heritage between them would have been lost. Changes would have had to be made to them individually. As symbols, however, changes could be made to any one object, and the change later propagated to all instances of the object, as a global update. Symbols carry the heritage relationship.

Walls could be replaced or new window units tried, first in one unit, then propagated to all or some subset of all the units. The symbol structure used in one student's project for a single unit is shown in Figure 6. Students learned to work with common elements, allocating them in complex ways to provide variety from a limited vocabulary.

The building compositions were also made into symbols. These also were replicated over the site, resulting in overall designs that had more than one hundred units. Students learned that one building composition could be juxtaposed to a different one with interesting results, as shown in Figures 7 and 8. Later, if design intentions involved changing one or more building compositions, its instances could be easily updated.
Design typically proceeds by adding detail, materials, color, detailing of structure, architectural finishes, etc. Such detailing, when followed consistently from the total project downwards, is called "top-down design".

Sometimes changes are not made at the bottom level, however, but within a composed component in the middle of the hierarchy. This type of change is easily accomplished, but requires discipline so as to not lose the heritage relations defined in the object hierarchy. A change is typically accomplished by exploding a symbol into its components and modifying some objects within it or their composition, then re-making it into a symbol and replacing the change everywhere the old symbol existed. In this way, a change is conceptually made in the middle or top of the hierarchy and propagated downward. The change is also propagated upward to the total design, of course.

These methods of design development provide a structured but highly flexible regime of design development actions. Other designers and teachers have also begun to explore hierarchical structures (Maderzo, 1990). The students spoke about this structure in terms of design vocabulary. They viewed their design process as defining a vocabulary of design elements and simultaneously creating a composition from them. Students quickly reached beyond simplistic repetition and explored ways to provide extra amenities, such as rooftop gardens and special views. With experience gained during the course, most students reached a high degree of competence in manipulating their design this way.

Working with the principles of object modeling constitutes a fundamental change in the design process. One has to resist the temptation of making a quick change of a line and instead act within a more disciplined process of isolating the change and encapsulating it as a symbol. Thus the immediacy of the intervention is replaced by a systematic procedure which, depending on the CAD system used, can be quick and easy or rather tedious. The spontaneous single intervention seems quicker, yet in the long run can never offer the structural advantages and the speed of exploration provided by object modeling.
Data Management Issues

These two strategies, based on object-based modeling, utilize many symbols, at many levels in the composition hierarchy. Objects can be nested quite deeply, as shown in Figure 9. One object may be part of several others and it also may be made up of others. When a change is made to a object, there is a need to regenerate all the affected objects. There is an obvious need to keep track of which object has which others nested within it.

In addition, as design proceeds, there are multiple versions of each symbol, as one is replaced with another. Memory management also led to multiple copies of symbols, as described below. Thus the names within the symbol are track over time and have multiple alternatives.

Traditional CAD systems provide no computerized help in keeping track of symbol versions and users must maintain extensive notes to keep track of their symbol or object structure. We recommended that students keep a notebook with them to record this structure.

After the class, it was recognized that an additional layer could be allocated and assigned to each symbol that could carry the symbol's composition. For example, if a window version was composed of frame version 7 and the mullions were version 3, these identifiers could be kept in a short list with the symbol. Attributes also could be used for this purpose.

Another issue was the size of the display files that were created over time. While the database remained relatively compact, because of the extensive reliance of symbols, the drawing files in CV, which carries information logically equivalent to a display list, grew very large. Manipulation would become extremely slow when the display list started getting swapped in virtual memory. If approached straightforwardly, each symbol has all the relevant design information within it: multiple 2-D projections at all levels of detail, 3-D model, and possibly material properties.

In Computervision, each symbol (in CV called a Figure) is stored on a separate file. A project is a directory and all its sub-files. Thus by splitting a unit symbol into several parts, for example making the inside walls a separate symbol from the outside walls, the outside walls could be loaded alone into the building and the city block complex, eliminating about two-thirds of the original project size.

In CV, it turned out to be very helpful to carry along multiple versions of all the symbols, each with different levels of complexity and size, as layers could not be unloaded from the display list (in CV, the drawing file). The different levels of complexity allowed one to only work with the information they actually needed. The cost, of course, is that one had to manage changes made to one version of the symbol to all the other versions. In many cases, this worked better than turning off layers, which in most systems does not unload them from the display list.

It is clear that CAD systems are still lacking in power and in certain dimensions of data management. One of these is the ability to manipulate large complex databases. Faster machines with more memory is one solution. But another limitation is the lack of object version control techniques. Needed are means to keep track of the component objects defining any other object. Also, needed are means to manage the version aspects of the different views of an object. These and other desired capabilities are described more fully in (Eastman, 1991).
**Evaluation**

The traditional means to evaluate studio work is a design review and jury. The faculty review of the student projects were impressed with the quantity of work students were able to generate in a ten week studio. The quick rendering capabilities were an obvious benefit. They also commented in a few cases on the spatial quality and level of visual resolution of the design development, which seemed an improvement over what the students had accomplished previously. The thought was that some students benefited from the high level of three dimensional feedback provided by the 3-D representation. The faculty who taught the course were not able to model the site in the system before the class (though this was intended) and negative comments were made about the lack of context for the student's work.

Our studio also encountered several issues of design presentation. Many faculty and students are engrossed by the subtle shading and quality of the renderings generated by high resolution displays and visualization software. But after this is accepted and internalized, we recognized that in many respects computer presentations of design projects are visually meager in comparison with traditional paper presentations. In a traditional jury, a student will hang five to ten different views of his project, with possibly the addition of a physical model. There is a visual richness that allows reviewers to quickly scan a lot of information, to review it in different presentations and scales. In contrast, CAD presentations on a screen present one or possibly four representations.

There is a need to provide the same amount of information with the same or greater richness in CAD presentations. Students should consider using multiple screens and composing multiple projections on screen. The benefits of paper presentation of a design is still strong. The media is not the message.

Early advocates of 3-D modeling suggested that it would soon replace 2-D for the representation of choice in architectural CAD. It has not done so, mostly for the reasons of cost, as an additional separate representation. The multiple representation approach to CAD allows the benefits of 3-D imaging to be integrated with only a small additional cost to normal drawing generation. The results can be distinctly different. It is a practical way for practitioners to take greater advantage of the capabilities of modern CAD systems. The cost of managing both 3-D and 2-D information on different layers is more than just doing 2-D layers alone, but much less than doing both separately.

Another issue emerged during the class regarding information coordination. As configured for the class, the CAD system provided the environment for defining the geometry. After it was generated, the geometry was transferred to the visualization machine. Here a designer would add color texture, lights and other visual details of the design, in support of rendering. The problem was that the design information is dispersed over multiple representations again. If energy analysis was integrated in, some material properties would be defined for visualization in the visualization environment, other material properties for the same materials would be added in the energy program. Any change made to the material properties would have to be remembered and updated to each representation separately. Some of the benefits of CAAD would have been lost.

Currently, the translator into Personal Visualizer assigns material classes according to layer. All the surfaces on a layer must be assigned the same material class. This use of layers is often at cross purposes with other uses. For example, it is usually not desired to put the interior walls of different rooms on different layers; in most cases, they will be
assigned common layers. But this forces them to take the same color or material in the visualization system. This led to much revision of layers in the later part of the course, in order to group objects according to materials. Many of the changes violated the earlier developed drawing conventions regarding which representations go on which layers.

To overcome this problem, we are building a material library within the CV and AutoCad environments that hold material properties as an attribute name. These material names will serve as cross references into multiple materials libraries associated with different translators. The unique visualization properties of that material will be automatically loaded when the CAD data is translated into the visualizer (Liggett and Jepson, 1991). Similar loading of energy related material properties will take place when the CAD data is run through the translator for energy analysis. Similar approaches will be used for other translators.

REFERENCES:


