THE INTEGRATION OF COMPUTER MODELING IN ARCHITECTURAL DESIGN

Leandro Madrazo
Swiss Federal Institute of Technology
Department of Architecture
8093-Zürich
(Switzerland)

ABSTRACT

The integration of computers in architectural design is explored from the perspective of both architectural education and professional practice. The main part of this paper attempts to define the conditions necessary for an effective interaction between computers and architects in the process of design. In the second part, a specific example, developed by the author during the course of his practice, is used to illustrate the use of available systems in professional practice.

GRAPHIC THINKING

The generation of graphic representations has always been an essential part of the process of design. Images and concepts originating in the mind of the architect are materialized through different graphic techniques. In turn, the visual representations provide new information that is evaluated by the architect. Thus, the design process is made up of continuous cycles of generation and evaluation of different sorts of visual outputs. This 'graphic thinking' has traditionally taken place with pencil and paper as the only tools. However, the graphic representations created with these basic tools are important to us because they reveal how we are thinking about a problem, not just what we think about it (Laseau, 1980). Similarly, in order to effectively use computers in design, we should first consider the way a computer allows us to represent 'how' we think about what we are designing.

The design logic of an architect facing a particular design problem can emerge as a result of creating graphic representations on the computer. An effective interaction of computer and designer should be based on the capabilities of the system to capture the knowledge the architect is generating during the course of the design. Therefore, design with an interactive computer system takes place on two levels: one based on the generation and interpretation of form, exclusive to humans; the other based upon the understanding of non-formal representations, which can be better handled by the computer.
Available systems fulfill these purposes at a very rudimentary level. They only provide standard procedures that the architect necessarily has to use, with no provisions for the perceptions of individual architects about their designs (Bijl, 1985). Moreover, most systems lack meaningful primitives and sets of operators that would allow the articulation of an architect’s design thinking. They make the architect fit into the particular restrictions of the package, something that has to do more with the limitations of programming techniques than with a particular design philosophy. In the worst cases, they stand at the same conceptual level as cardboard model making and provide no new conceptual mechanisms for addressing design.

Solid modeling can be considered an exception to this scenario. In solid modeling, the generation of form occurs within the context of a well-defined set of operators acting on a closed universe of geometric elements. A similar approach adjusted to the purposes of architectural design could provide the appropriate framework for overcoming the limitations of the available systems.

THE UNDERLYING STRUCTURE OF FORM

Architectural form can be obtained with a solid modeling program in much the same way as any generic form. For example, in order to create a 'frame' we can subtract openings from an initial rectangular prism (Fig. 1).

![Fig. 1- A frame as a result of a subtraction](image1)

or we can start with the vertical and horizontal solids (columns and beams) to get a similar result with a union operation (Fig. 2).

![Fig. 2- A frame as a result of a union operation](image2)
Although the results of both methods may seem similar, a difference already exists the moment that the first elements and operations are selected. This difference appears clearly when we create new frames by changing the parameters of the initial elements. In the case of the frame produced by subtracting openings, for example, other versions of the frame can be generated by changing the size of the opening (Fig. 3).

![Fig. 3 - Different frames are generated as a result of changing the size of the opening](image)

Similarly, the transformation of the frame created by the assembly of vertical and horizontal solids takes place by modifying the parameters that define these elements (Fig. 4). In the first case, the opening can be called the *active design element*, that is to say, the element whose manipulation results in new versions of the frame. In the second case, columns and beams become the 'active design elements'.

![Fig. 4 - Different frames are generated as a result of changing the size of the parts](image)

What this comparison illustrates is that an understanding of the underlying structure of the form is already present in the form itself. We are already expressing 'how' we think about the frame when we build the computer model. Incidentally, the use of punched openings at this level of abstraction could be a reference to concrete construction, while the assembly of separate parts could refer to steel construction techniques.

The first definition of the frame can be questioned when new interpretations necessarily emerge during the process of design. For example, the regular frame can be read as having an outer frame as a separate part (Fig. 5). As a result, the outer frame might become an 'active design element', in turn creating new types of frames.
Changes that are performed at the graphic level by the designer have other implications for the system. Elements are redefined and new hierarchies created as a result of those changes. As a result of the interaction with the computer, the architect is freed from the technicalities of building form with the computer (which are still overwhelming and nonintuitive in many packages) and can concentrate on specific design decisions. Particularly, the exploration of multiples alternatives within the same topology would seem natural to an architect interacting with such a computer system.

DEFINING OPERATORS

The example of the generation of a frame can be understood in terms of the transformation of a geometry into a structured object (Fig. 6). The resulting object inherits the conceptual structure that is implicit in the operator. As in the example of the frames generated by subtracting openings or as the assembly of parts, each 'structured object' is different, although the 'geometry' of the resulting object is equivalent.

In this context, new versions of the frame do not necessarily have to be the result of direct manipulation of formal elements. Since the system understands the structured object as being made up of an initial geometry with an operator applied to it, new frames can be made by changing either of the two components. For example, the same operator can be applied to a different geometry (Fig. 7)
or a change in the parameters of the operator result in a new frame (Fig. 8).

In theory, the architect could act at the level of graphic representations leaving the system to interpret the meaning of the formal transformations he is performing. Thus, the system deduces the operator by comparing a geometry to a structured object (Fig. 9).
During the process of design, other operators will be devised. For example, at some point the enclosure of a building can be understood as the repetition of the same component. The transformation from the low-level component into a higher one can be defined by an operator (Fig. 10). Also, the same operator can be applied later to generate other types of components, such as columns.

![OPERATOR 'A'](image1)

![OPERATOR 'A'](image2)

Fig. 10 - The same operator can be applied to generate different architectural elements

In the case of repetitive patterns, such as the ones considered in these examples, it is most straightforward to have direct access to the parameters of these patterns in order to get different formal solutions. However, the challenge for the architect lies in the discovery of less evident and more idiosyncratic operators which can then be formalized and understood by the computer.

MULTIPLE REPRESENTATIONS

Developing a design by moving back and forth between a rough formal representation and detailed studies of the parts is one of the most intuitive strategies adopted by architects. This strategy reflects the need to isolate different parts from the overall complexity of a design problem in order to handle them properly. It also provides the opportunity to gain a thorough understanding of the problem by exploring quickly multiple paths before putting too much effort into defining one possible solution. Finally, it paves the way for the emergence of a solution that synthesizes all of the different aspects considered during the process.
If we accept this strategy as essential to the architect, an interactive system should provide the ability to work with multiple representations of architectural elements. The need for multiple representations has already been pointed out as a way to solve the inconsistencies between computer models and traditional drawing conventions (Eastman, 1987). In a broader sense, every representation is the expression of an isolated set of design issues extracted from the overall design problem. Thus, the significance of a plan lies not so much in its being the result of sectioning space through a plane but rather in its being one type of representation that allows one to consider a specific set of issues which it is not possible to evaluate in other views (for instance, circulation and areas).

Returning to the example of the frame, we can consider three different representations of the frame made up of separate pieces (Fig. 11). Each representation shows a different level of abstraction. Together they could represent three different stages in a chain of design decisions (although not necessarily following the proposed order).

![Fig. 11 - Three different representations of a frame with different levels of detail](image)

In 'level 1' the members of the frame are represented by single lines. At this level of abstraction we can get information about dimensions of the span and the number of porticos of the frame, for example. The representation in 'level 2' allows other issues to be considered, such as the proportion of void to solid. In 'level 3', the use of I-shaped sections brings up other kinds of design issues, for instance, what is the orientation of the I-column with regard to the beam.

Each representation allows us to deal more effectively with a separate set of design issues. An architect designing with the computer will find it necessary to work with the level of representation that reflects the issues he is considering at that point in the process. It should therefore be possible to move back to 'level 1' and change the number of spans from three to two, for example (Fig. 12). This change could be performed at the level of representation where only the distances between the center lines of the bars are depicted (the orientation of the I-shape columns is not relevant when the design issue at stake is the change from three spans to two). A link would allow us to go back to 'level 3' and see the new version with the corresponding level of detail.
This sequence of design decisions would not necessarily have to start at 'level 1'. It should also be possible to modify the proportions of solid and voids at 'level 2' and expect that the representation at 'level 3' reflects the changes.

Current systems do not facilitate the simultaneous work with multiple representations of the same architectural element. Because of this, an architect working with these systems is compelled to understand design as an incremental process (Madrazo, 1989). This represents a problem when it is necessary to step back and reconsider decisions made at earlier stages in the design process.

Every representation provides information about what is known about the design at a given point in the process. The decision to proceed in a certain direction is very much based on the information that those representations provide. For that reason, an environment that facilitates going back and forth through different levels of abstraction opens a new range of possibilities for an architect (Liggett, Mitchell, Tan, 1990).

DISCOVERING THE RULES

The definition of operators relies on the ability of each individual architect to formalize his design logic for a given design problem. Shape grammars could provide the conceptual instrument needed to formalize the knowledge that an architect acquires during the design process. However, shape grammars have become basically an instrument of analysis with the specific purpose of capturing knowledge about a coherent body of architectural examples. Once the knowledge is encoded, other designs that comply with the formalized rules can be generated.

In the context of intuitive design, shape grammars could acquire a different character. The process does not necessarily have to start with the formal declaration of rules. An architect might rely on his first intuitions to start up the process in a certain direction. At some point in the process, emerging 'rules' can be made explicit and from there the design can continue making extensive use of the rules. However, the application of shape grammars in this context would not be a guarantee for good design (Schmitt, 1988).
Fig. 13: Formal Games: from intuitive creation to formalization of rules.
(extracts from an animated form generation, L.Madrezo, ETH Zürich, 1990)
The example in Fig. 13 began as an intuitively generated object, without conveying to a previous set of rules. As soon the object acquired some identity, it became clear that the generation of form could be formalized as the rule in Fig. 14. The rule consists of four basic steps. A face is first selected from an existing form. In the second step, following the selection of a point, the face is divided into four areas (one or two areas are possible if the selected point is on a corner or an edge). The third step asks for the selection of the area which is to be extruded (positively or negatively) in step four.

The initial purpose for these 'Formal Games' is to isolate the generation of form from any particular architectural constraint (corridor, entrance, facade, light). The hypothesis is, that no matter how generic the form is, the adaptation to a particular architectural constraint is always possible. This conscious perversion of commonly accepted design practices can be used to illustrate other sorts of limitations in the implementation of computer techniques to assist design.

For example, it is often taken for granted that the organization of space is synonymous with rectilinear arrangements in plan view. An idea for a particular space could, however, originate from sources that do not necessarily need to be represented by either rectangular arrangements or plan views. It is true that in the end any design will have to be represented with plans and that some of them could be described as an arrangement of rectangles. What is relevant for the creative process is which 'conceptual mechanisms' we use at the beginning of the process. The way we think about a design issue already says a great deal about the solutions we will be generating.

A plan view of one of the stages of growth of these 'Formal Games' is shown in Figure 15. Since the forms were always generated in axonometric view, the resulting plan reflects little consideration for issues such as adequate dimensions or functional adequacy. At the moment that we start seeing this representation as the layout of a house, for example, it will acquire a different meaning. The abstract lines will conform to the notions we associate to the architectural constraint 'house'. The design goal will then be to reconcile those notions to the abstract vocabulary already defined.
For the purposes of architectural education, these 'perversions' of the design process can serve as a pedagogic technique. In order to achieve the integration of computer technology in the architectural curricula, it becomes crucial to define the scope within which the computer can add new dimensions to traditional design thinking. Detaching and developing specific design issues opens new opportunities for design exploration in a context unique to computers.

THE COMPUTER AS A DESIGN TOOL IN ARCHITECTURAL PRACTICE

The development of computer-aided design in architecture seems to be taking two separate paths: one that conforms to the requirements of professional practice as they exist and a second one that explores the impact of computer technology on the way we conceive and make buildings. While the goals in the first case can be more easily described and handled in a professional environment, the second is still a matter of intellectual speculation confined to the academic environment.

The ideas described in the previous pages focus on the generation of form and are addressed to the integration of computers in architectural education. In professional practice, however, isolating specific design issues is not a sufficient method for addressing the whole complexity of architectural design. Thus, the integration of computers for design in architectural offices brings up other kinds of issues.

In order to illustrate a current approach in using computers to design in professional practice, I will describe the process followed during the development of a project I worked on during my time at the Los Angeles office of Skidmore, Owings & Merrill using the AES system.

A CASE STUDY: A COMPETITION IN CLEVELAND, OHIO

The site for the project borders the river and is surrounded by an industrial area. The program asks for a mixed-use complex with office, residential and retail space. The project has to comply with existing master plan regulations as well as to integrate with the urban conditions of that area of the city (grid, neighboring structures).

References to the industrial character of the area are already present in the first sketches. For example, the bridge that connects the office tower to the housing on the river is an allusion to the nearby steel bridges. At this early stage of the process sketching becomes an essential tool in the search for images and analogies. From the sketches, the design continues using cardboard models as well as computer models.

The implementation in the computer of the ideas represented in the sketches has necessarily to go through an analytical process. Consciously, the sketches try to be as explicit as possible about the different components of the project: bridge, tower, top of the tower, housing. Therefore, sketching provide not only for the synthesis of the ideas but also for starting the analytical process that is to be continued on the computer (Fig. 16-17).
The computer model gets built right from the sketches, no other hand drawings are made for that purpose. The separate components are modeled directly in three dimensions. Two-dimensional projections are used only as a tools for building the 3-D model. As a result, the creation of form and space acquires a different character than it would have had if the model had been built from plans and elevations previously laid out. It becomes essential to be aware of the boundaries of every element, their level of detail, hierarchies, and ways in which the parts can be assembled. Most important for the architect, however, is the feeling of working directly with form in space, liberated from restrictions such as fixed views.

The first ideas underwent different transformations during the design period (Figures 18,19). Even though changes were made in the formal vocabulary, the basic structure of the project remained unmodified. For example, the top of the tower is still a distinctive part although it is formally different than in the earlier versions. What this reveals is that some important aspects of the design are independent from a particular formal expression. A first goal for the implementation of the design in the computer is to capture that underlying structure of the design.
The mechanisms that CAD systems provide to capture this underlying structure are specific to computer technology. Working with today's systems, it is necessary to make the transition between architectural and computer concepts. Typically, different parts of the building are broken down into separate files, elements with comparable levels of detail are stored in symbol libraries and hierarchy structures built through nested symbols (Madrazo[1], 1989).

CONCLUSIONS

An effective integration of computers in design has to first go through an understanding of the role of the generation of form in the process of design. Other important aspects, which are also part of architectural design (programmatic, functional), have been purposely avoided in this paper. Though they have not been mentioned, it is assumed that they define the boundaries within which the activity of the architect takes place.

Traditionally, the isolation of some issues from the overall complexity of architectural design has been a necessary step for the implementation of computer techniques in design. In some cases, these techniques have focused on the issues that are well-suited for computer implementation, for example, the automatic generation of plan layouts. However, it is not clear that by providing the means to generate larger number of solutions in less time we are giving the architect a tool for better design. A more challenging path for future development is to address the issue of the generation of form with the computer.

Architects need to be able to work on the computer as intuitively as they do with other media. Thus, free exploration of formal solutions is a necessary interface between traditional design thinking and the new possibilities of a computer tool.

In order to achieve an effective integration of computers in architectural design, it will be necessary not only to provide an architect with the adequate tools but, more important, to transform the design logic that architects have built-up over time. The impact of the integration of computers in architectural design can be evaluated only when the new conceptual mechanisms provided by computers have transformed the way we conceive architecture.

ACKNOWLEDGMENTS

The author would like to thank Chen-Cheng Chen, Dina Cheyette, Sharon Refvem and Gerhard Schmitt for their suggestions; and to the Los Angeles office of Skidmore, Owings & Merrill for letting me use the graphic material.
REFERENCES.


