The Semantic Modeling Extension (SME) prototype implements a unique approach to integrated architectural CAD that places the drawing act first in the design process. After drawing a design idea using a computer-aided system, the designer interprets the design, providing semantic content to the graphic entities. An interpretation expresses the meaning of the design with respect to a particular issue, such as structural efficiency, energy consumption, or requirements for egress, and provides reasoning to evaluate the design addressing that issue. A design may have many interpretations to express the multiple issues that are relevant in a design project. The designer may add or delete interpretations of the design as issues change during the course of the project.

Underlying the SME prototype are the concepts of form, function, and behavior. In the prototype, evaluation of a design is done by deriving behavior from the graphically represented forms and relating the behavior to stated functions or requirements.

The concepts of interpretations and form, function, and behavior together establish a virtual product model for design. In contrast to component-based approaches to product modeling that tightly bind form representations to their behavior and function, a virtual product model allows the designer to manipulate the relations among these three descriptors of a design, and thus manipulate the semantics of the design entities. By distinguishing between the act of proposing a design by drawing the conceived form and the act of assigning meaning to the form, the virtual product model approach supports both graphic thinking for design synthesis and symbolic reasoning for design evaluation.

This paper presents a scenario of the use of the SME prototype in building design; provides an analysis of the design process and computational support described in the scenario; contrasts a virtual product model approach with a component-oriented product model approach; describes the software implementation of SME; and presents implications and conclusions regarding design process and technical integration.

Scenario

The following scenario illustrates the concepts behind SME with an example of the software's use in the design of a suite of rooms for a hospital. Although the scenario is a laboratory experiment, the situation and activities are built upon observations of an actual recently-completed building project executed with manual, conventional design methods. The scenario begins with a top-level objective: define spaces to accommodate equipment (a cyclotron) that will provide radioisotopes for a radiology lab. The hospital administrator has provided to the architect a very rough estimate of the size of the new hospital addition in terms of overall floor area, approximate budget, and a time frame for the project. The architect has also provided the architect with contacts at the cyclotron manufacturer and brochures describing the equipment.

Instances of negative effects of computer applications, the latter conclusion would appear more plausible. Clearly the health of the profession is dependent on a broad set of conditions. More to the point, our vision of the use of computers has been constrained by our vision of the profession of architecture. Machines become what we think they are. Colin Forbes of Pentagram adds “The quality of
Initial 3D Model

The architect’s first step in producing a design is to sketch ideas for the new addition using a 3D CAD system. The architect lays out a basic volume to accommodate the cyclotron and the radiology lab and then draws walls, floors and roofs by defining solid entities such as boxes and extrusions. Windows and doorways are cut into the walls using subtraction operations on the solids. Figure 1 illustrates this initial model. As the drawing is represented in a 3D modeling environment, the designer may obtain any perspective projections desired and may render the model as needed to convey the ideas to the client.

![Initial sketch of the cyclotron room for a hospital. The designer draws a design using 3D CAD.](image)

Spatial Interpretation

Having obtained spatial evaluation software from the cyclotron manufacturer, the architect loads this critiquing module into the design system using the Interpretation Manager shown in Figure 2. Clicking on the New button, the architect creates a "spaces" interpretation. At this point, the solids drawn to represent the elements in the design have no explicitly expressed semantics, and the evaluation software is as yet incapable of providing automated reasoning about this particular design.

![Interpretation Manager provides commands for loading new interpretations into the design environment. Here, the designer is about to load a spaces interpretation.](image)

Using the Interpretation Editor dialog box shown in Figure 3, the architect interactively annotates relevant graphic form entities with their semantics. By clicking on the New button, the architect can designate a new feature of the design within the spaces interpretation. The architect indicates which class of feature to create, the feature name, and what graphic entity to associate with this feature. The architect draws volumes to represent the individual rooms and identifies each room within the spaces interpretation, such as the cyclotron room, the radiology lab, and the water recirculation system room. By providing a list of relevant objects, the Interpretation Editor guides the designer in specifying the semantic content of the drawing entities.

The geometric attributes of features may be adjusted by editing the graphic model, while the semantic content may be changed by changing the feature class or its attribute values. At any time, the user can associate a graphic entity with a different feature. For instance, a storeroom may initially be designated as a custodial store but then, because of

Howard's End

WHAT YOU SEE IS WHAT YOU GET

ACADIA 1994
proximity to the cyclotron room, be redefined as equipment. The designer can control the graphic display of the design by hiding, showing, or highlighting the graphic entities that correspond to particular features.

Figure 3: Editing the spaces interpretation. The designer interactively annotates relevant graphic form entities with semantics within the spaces interpretation.

Egress, Construction and Energy Interpretations

To check that the building satisfies code requirements for egress, the architect loads an egress evaluation module. Using the Interpretation Editor, the architect identifies the spaces and doors that are relevant to egress, as shown in Figure 4.

The hospital administrator also needs a preliminary cost estimate and a construction schedule for the new wing. Using the Interpretation Manager, the architect loads a construction cost and scheduling evaluation module into the CAD system. The architect draws the footings, HVAC units, lighting systems and equipment so that the physical components of the building are all shown as solid objects. Using the Interpretation Editor, the architect designates construction features, as shown in Figure 5. The construction features refer to items in an assembly-based estimating database such as spread footings, block walls, and drywall partitions.

To address a fifth issue, that of energy consumption of the building, the architect loads another evaluation module and once again identifies the features of the building, this time considering energy use. These include the exterior walls, the windows, the roof, the lighting systems, and other feature classes. Most, if not all, of these features have already been drawn; they need only be annotated as features in the energy interpretation and assigned appropriate semantic values.

Figure 4: Editing the egress interpretation. The designer interactively annotates graphic form entities relevant to egress.

Design Evaluation

Having annotated the graphic model with semantics within each interpretation, the architect invokes automated evaluations of the design for each of the issues. For each interpretation, the design's performance is predicted and then compared to the design requirements of the relevant issue. In a few seconds, the cyclotron reasoning module evaluates the sufficiency of the design in accommodating a cyclotron room and a radiology lab. It examines the dimensions of the rooms, vertical clearances, accessibility for installation, maintenance and operational requirements, and adjacency of related spaces. Another reasoning module examines the size and connectivity of rooms and doors expressed in the

what we do with technology depends on what we think we are doing. It depends on what we are aiming for; how we see our contribution. In other words, "what we see is what we get."

Until recently we too often saw the profession through Howard Roark's eyes: aloof, egocentric, self-sufficient. Architecture was about being an architect-creator, suffering the realities of clients and budgets. Dana Cuff has pointed out
egress interpretation to test whether the design meets the relevant code provisions for egress. A construction evaluation tool provides a cost estimate and a preliminary construction schedule. The architect also types in values for the cost budget and the project time constraints that were provided by the administrator, and the evaluation tool compares the predicted cost and schedule to these requirements. Upon designation of the climate for the project and a set of energy consumption objectives, another software module evaluates the design regarding its predicted consumption of energy.

![Figure 5: Editing the construction interpretation. The designer interactively annotates graphic form entities relevant to construction.](image)

Each automated evaluation tool provides an explanation of its results and suggestions for improvements. This information is linked to the 3D CAD model so that the user may display a textual explanation and highlight the affected objects within the context of the actual design. Figure 6 illustrates the results generated by the energy evaluation that a window should be shaded.

The evaluation process leads to a set of suggestions about design problems. The architect chooses one or more problems to resolve, changes the form appropriately, and then invokes the evaluation process again to check that the introduced change has the desired effect and to identify any unintended adverse consequences of the change. Changes to the design may include drawing new objects, modifying old objects, and re-interpreting the semantic values of graphic entities. Most of the work is in editing the form of the building using the graphic editing tools.

![Figure 6: Results of the energy evaluation. Here, a suggestion regarding the energy performance of a window is shown.](image)

The designer may query the design model to retrieve a list of interpretations that affect a particular graphic entity to identify which issues would be affected by a change to the entity. By intersecting or subtracting two interpretations, the architect can focus upon the interactions between two issues.

Presentation
The process has taken perhaps an hour, including some iterative design of the cyclotron facilities and experimentation with different wall constructions to study their effects upon energy consumption, construction cost and schedule. The architect now has preliminary descriptions of the building’s appearance, spatial layout, code conformance, construction cost, construction schedule and energy consumption that can be shared with the client. Each of the descriptions is in the form of an evaluation result expressed as an annotation of the 3D graphic model.

"I don't create architecture in order to have clients, I have clients in order to create architecture."

Howard Roark
The architect and hospital administrator examine the issues together. The administrator expresses some small concern with the construction cost, but larger concern with the operating cost. She suggests brick exterior walls to provide a more pleasant, less utilitarian image for the new wing. In addition, she informs the architect that the hospital has begun considering the purchase of the cyclotron from another manufacturer, and that she needs to compare the fully installed costs for each cyclotron option. She would like to study a design that allows the hospital to defer the selection of the cyclotron until after completion of the building.

Iteration with New Information
The architect refines the design based upon the new instructions. After obtaining another evaluation tool from the second cyclotron manufacturer, perhaps by downloading it across a wide-area network, the architect loads it into the CAD environment. He adjusts the room dimensions and access routes and builds a new interpretation to address the new issue of the second cyclotron manufacturer. In hopes of reducing operating costs, the architect inserts skylights in the lab space to provide natural lighting and reduce lighting-based energy consumption. He adds these new graphic entities to the construction and energy interpretations. The evaluations of each alternative supplied by the evaluation software provide guidance toward improving performance as well as a documented trail of the evolution of the design. By carefully adjusting the dimensions of the cyclotron room and lab space, the architect achieves a design that satisfies both cyclotron alternatives. At the end of the day, the architect has studied several design alternatives, and has produced accurate comparative evaluations of the alternatives and convincing visualizations.

Analysis of the Scenario
The scenario describes a believable design process and computer tools that support that process. The SME prototype provides a distinctive and appropriate environment for the early conceptual stages of a building design project by:

- addressing multiple issues in design;
- accommodating changing information needs as a project progresses; and
- integrating graphic thinking and symbolic reasoning, allowing for free consideration of the design in both modes.

Design Evaluation
It is widely held that design consists of an iterative cycle of activities in which a design idea is formulated in response to requirements and tested for sufficiency to those requirements. These activities are typically defined as analysis, in which the design requirements are formulated from a study of a problematic situation, synthesis, in which alternative solutions to the problem are conceived and documented, and evaluation, in which the solutions are tested for predicted performance and judged for suitability and optimality [Asimow 1962]. The scenario describes a process in which the designer rapidly iterates through the analysis-synthesis-evaluation cycle. As a result, the process described in the scenario is significantly accelerated over non-computerized methods, allowing the designer to consider more alternatives. Due to delays in scheduling consultations with specialists and the time required for often tedious and repetitive calculations, evaluation is often a bottleneck in the design cycle. Integrated, automated evaluation software may drastically reduce the time necessary for considering multiple design options during preliminary design, from a prohibitively long period of several days to an inconsequential few minutes. SME focuses computational support on the evaluation stage. The computerized evaluation tools also provide more rigorous and more consistent evaluations and higher quality documentation of the performance of alternatives than is generally prepared using manual or non-integrated methods.

The SME prototype offers little support for the analysis and synthesis stages of the design cycle. The functional requirements resulting from analysis are represented to some degree by the collection of interpretations that are active in the system. The provision of 3D graphic editors allows the representation of the results of synthesis. In the

that "Howard Roark in The Fountainhead is the epitome of the solitary, uncompromising architect. He abhors the social process that architecture engenders:. The movie glamorizes the myth that all collaboration leads to mediocrity."3 This sort of architect, interested in self-expression rather than problem solving, has little use for the information and communication that
scenario, however, the actual synthesis process is carried out solely in the mind of the architect.

Multiple Issues

    Design, especially during the early stages, involves the consideration of multiple issues and their integration into a balanced whole. As revealed by protocols of designers at work, architects alternate rapidly among contexts of reasoning to achieve a balance [Schon 1983]. Each context addresses a particular design issue. Within a context, the designer employs a vocabulary to name the things, often called features, that are important. A feature is commonly defined in the mechanical engineering profession as any geometric form or entity whose presence or dimensions are germane to functional reasoning [Dixon 1986]. Schon has observed that the vocabularies used in different contexts for naming features overlap and that particular names have multiple meanings across contexts. He comments that architects endow their work with subtle shades of meaning from these ambiguities, drawing inspiration and achieving complex integration of function.

    The concept of interpretations establishes a framework for accommodating multiple issues. The scenario focuses upon five design issues for the hospital addition: the spatial sufficiency of the suite of rooms in accommodating a cyclotron, the provision of egress in conformance with building codes, the projected construction cost, the construction schedule, and the energy consumption and resulting operating cost. In addition, some consideration of aesthetics is apparent in the decisions regarding exterior wall construction. Each issue defines functional requirements for the project. The five issues are supported by four interpretations: a spatial interpretation, an egress interpretation, a construction interpretation, and an energy interpretation. Using its list of available feature classes, each interpretation establishes a vocabulary that expresses the semantic context of a design within a particular issue. Each vocabulary is independent of the vocabularies of other interpretations, allowing designers to use natural feature names rather than forcing agreement upon unique names for every concept.

In the scenario, the architect considered interactions among the various issues. SME provides some support for studying such interactions by hiding or showing graphic entities that are shared among interpretations. A graphic entity can also report a list of interpretations that reference it. This functionality is a powerful tool for studying the interdisciplinary effects of design decisions.

Changing Issues

    At the early stages of a project, the design issues are in the process of being established. Juggling budgets and schedules, the owner may change objectives for the project or defer some objectives to a later project. The architect is on the lookout for both emerging opportunities and the client’s unstated objectives. In the scenario, the hospital administrator adjusts the functional requirements of the building addition by instructing that it accommodate a cyclotron from either of two manufacturers. This change imposes new requirements for spatial allocation and access. Design tools for the early stages of a project must accommodate changing issues and new information needs and relations. This requirement has been explored in design database research and has been referred to as “dynamic schema extension” [Eastman 1981]. To accommodate changes to informational needs, the SME prototype provides for dynamic loading of evaluation modules by the end user during a design session. To consider a new issue, such as the cyclotron by a second manufacturer, the architect need only load the new evaluative module and create an interpretation for it by associating features with particular graphic entities in the CAD model.

Graphic Thinking and Symbolic Reasoning

    A salient characteristic of this scenario as a design process is its fluid integration of tools for graphic thinking and for symbolic reasoning. The notion of graphic thinking has been well established in design methodology [Crowe 1984][Lasen 1989]. The graphic portrayal of design ideas is an invaluable tool for comprehending situations and consequences and exploiting emergent opportunities.

Howard’s End

    ...a professional body of knowledge remains incomplete...

ACADIA 1994
Computational methods of reasoning, however, depend upon symbolic methods. In architecture, numerous prototypes and commercial systems implement symbolic reasoning that assists in design tasks. Using predominantly procedural methods, DOE-2 and Solar 5 provide design evaluation in the area of energy [Birdsell 1990] [Milne 1993]. Expert systems such as HI-RISE for structural design and ICADS for integrated architectural design provide examples of rule-based methods applied to architectural evaluation [Maher 1985] [Pohl 1990].

The SME prototype integrates tools for both graphic thinking and symbolic reasoning. In our scenario, the architect first uses graphic thinking to compose a design synthesis. Evaluation is delayed to allow the emergence of design ideas within the graphical environment. After a design idea has been portrayed, the architect invokes automated reasoning tools that use symbolic methods to evaluate the design. The architect can move freely between the graphical representations that facilitate internal graphic thinking and the symbolic representations that externalize reasoning in computer software. The interpretations provide the linkage between these two reasoning modes. They map a graphic representation of the design into symbolic representations.

Although our scenario emphasizes a design cycle in which drawing comes first, the SME prototype allows the designer to begin the design process with symbolic representations. For instance, the first step in the design process could be to enumerate a list of spatial features and adjacencies that must be satisfied by the design. This list would be represented as an interpretation that has not yet been mapped to a graphic model. After completion of the symbolic spatial representation, the architect can create graphic representations and link the graphic entities to the features. Of course, an architect need not complete the symbolic representation before beginning the graphic representation, but using SME can alternate freely between both modes of representation.

A Virtual Product Model

Like a conventional product model, SME addresses the goal of support for information exchange among multiple applications. Unlike a conventional product model, SME does not employ a central, comprehensive, product representation defining complex interrelationships among components, created by software developers. Instead, SME is based upon a concept of distinguished form, function and behavior that are related by the end user during the process of design. SME is an implementation of a virtual product model that allows the end user to define and redefine a product model during the course of a project.

Form

In SME, form is defined primarily as the geometry of the design. Form is what architects draw in plan, section, elevation and perspective. Form is often physical, such as a wall or column, but need not be, such as a space or region. Form is naturally represented and manipulated using computer graphics techniques. Examples of form entities are planes, boxes and complex 3D shapes. Forms are the result of the synthesis step in Asimow's design cycle and "follow" from function. In the scenario, form is best understood simply by examining the graphic representations of the building.

Function

Synonyms for function include requirements, needs, intents, and objectives. Functions express desires for the building: the client desires a space to accommodate a cyclotron, the client desires a pleasing exterior finish, or the client desires construction within a stated budget. Functions are the result of the analysis step in Asimow's design cycle and originate from a problematic situation.

Each of the issues addressed in the scenario groups different functions. The spatial issue focuses upon providing the needed spaces, adjacencies and access routes for installing and operating a cyclotron to support medical uses of radiotopes. It leads to requirements about minimum sizes of doors, access between rooms, adjacency of one room to another, electronic technologies offer. To the extent that such architects practice in isolation, disconnected from each other, the formation of a professional body of knowledge remains incomplete. In the context of such limitations, a major benefit of computers, high quality information, is denied. Without shared theories of design process, appropriate delegation of tasks to computers is thwarted.
and dimensional requirements for spaces. These functions are implicit in the choice of the cyclotron and are not directly controllable by the architect. The cost issue establishes a monetary budget for the construction cost as its function. The schedule issue establishes a temporal budget as a function. The function expressed by the energy issue is to achieve thermal comfort at acceptable levels of energy consumption. It is elaborated into functions specialized for the particular climate of the project, such as reduction of unwanted solar gains through windows or reduction of thermal gains produced by electric lighting.

Behavior

Behavior is the expected performance of the design within a particular situation in a particular reasoning context. Behavior is typically derived from the form model by some kind of prediction. Behaviors must map in some well-defined way to the functions in order to support evaluation of the design. Consequently, much of the research literature uses the two terms interchangeably. However, the distinction between desired performance (function) and predicted performance (behavior) has been established by other writers [Luth 1991] [Flemming 1992]. Behaviors are the consequence of design forms. Each behavior is responsible for predicting some single performance of the design with respect to one or more forms.

Behaviors may be very simple, such as the area of a form, or much more complex and subtle, such as the energy flow through a form in a particular climate at various time intervals. Each issue in the scenario is concerned with a particular collection of behaviors. The spatial issue is concerned with areas and adjacencies of rooms, connectivity of doors to rooms, and sizes of doors. The cost issue is concerned with the quantities of various elements. The scheduling issue defines construction activities and construction dates as behaviors of the form. The energy issue is concerned with convective, conductive and solar radiation energy flows through various features.

Several behaviors have been defined for the evaluation tools in the SME prototype. Length is the larger of the two horizontal dimensions of the bounding box of a solid. Width is the smaller of the two. Height is the vertical dimension of the bounding box. Volume is computable for any solid. Area is defined with respect to various planes. Connectivity is a behavior that serves as a base class to both a supported-by behavior and a door-room connectivity. These basic geometric and topological behaviors form the foundation for the more complex, issue-centered behaviors described above.

Interpretations

In the virtual product model implemented by SME, form, function and behavior are all represented as distinct objects. The glue that associates them to allow design reasoning is the interpretation. An interpretation defines a system of reasoning about a design to address a particular issue. It collects functions and behaviors and provides a means of assessing whether the expected performance satisfies the desired performance. Through the mechanism of feature creation, an interpretation provides a method of associating a designed form representation with behaviors and functions.

Interpretations collect evaluative reasoning according to a particular issue. By selecting among interpretations, the building designer identifies the relevant issues for the design project and acquires automated reasoning support for those issues.

Features

Each interpretation defines a set of features that are relevant to the issue addressed by the interpretation. A feature is the means by which the system relates form, function and behavior. By interpreting a graphic entity as a particular feature, the architect associates the form represented graphically with behaviors that can be used to satisfy particular functions.

Interpretation Manager

Support for dynamic schema extension is achieved using an Interpretation Manager. This software object maintains the list of loaded interpretations, allowing the end user to add new interpretations and to delete interpretations that are
no longer needed. It also supports functions that range across interpretations. Implemented in the prototype are capabilities for union, intersection and subtraction operations on the feature lists of interpretations and query of a graphic entity for its membership among multiple interpretations. These commands are useful for studying interactions of issues in the design.

The interpretation manager provides a concise interface by which virtual product modeling capabilities may be integrated with a computer graphic system, such as a commercial CAD system. Automated design evaluation modules could be connected to any CAD system that provides an interpretation interface.

Related Research

This research is a distinctive approach to the definition of design semantics and how to represent them to produce useful and effective software for preliminary architectural design. In the commercial world, representation of design semantics has generally been achieved by associating meanings with drawing layers. Researchers have generally adopted a component-oriented approach for development of integrated CAD prototypes. Virtual product modeling is a unique approach to the representation of design semantics that provides a clear method for achieving a flexible design software environment and modular design reasoning.

Layer Paradigm

Virtually all commercial CAD systems provide the ability to assign graphic entities to layers, analogous to the overlays in pin-bar drafting. As layers can be made visible or invisible, the user can control the visual portrayal of the drawing by toggling layers on and off and moving entities from one layer to another. Layers provide a means for collecting entities with related meanings, often by professional discipline, such as all structural elements onto one layer, all architectural elements onto another, and all mechanical elements onto a third layer.

Unfortunately, layers have two critical limitations. A layer name can only provide a semantic identifier to a collection of entities, perhaps the context by which the entities may be considered. The semantic content of individual entities on the layer cannot be represented by merely the layering scheme. The use of complex primitives, known variously as blocks, cells, or symbols, provides a partial solution to this problem. A more serious limitation is that layer membership is treated as an attribute of the entity. An entity may have one and only one layer assignment. Thus, if meaning is encoded into layers, an entity may have one and only one meaning within the design. Building designs can rarely be decomposed into collections of entities with single, distinct meanings. For example, a wall may appropriately be included on the architectural, the structural and the mechanical layers. Many workarounds have been devised for addressing this limitation, but none of them is fully satisfying. The layer paradigm, while appropriate for automating overlay drafting, is inadequate for representing buildings in a semantically rich and complete way. Nevertheless, layer-based representations of building semantics are widely used in commercial CAD tools, such as ASG and LightCAD [ASG 1992][EPRI 1992].

Component Paradigm

The layer paradigm has been widely rejected by researchers, generally in favor of a component-oriented approach. In this view, a building is decomposable into its parts, each of which has pre-determined semantic content. Research using a component approach focuses on the correct and comprehensive definition of components so that they may support a variety of design evaluations. Attention is devoted to the development of specialization (kind-of or is-a) hierarchies and part-of hierarchies and to the definition of attributes and relations among classes and instances. Much of this research blends implementation methods of relational databases, object-orientation, and frame-based semantic networks. Many projects have proposed

However, there is reason to be optimistic about the future of architects and the potential contributions of computers to architecture. We have finally come to Howard’s end the end of professional isolation, the crisis model of design management, seat-of-the-pants processes, the secret craftsmen’s guild mentality and marketing myopia. The Fountainhead has revealed itself as paper-thin and laughable. But it will not suffice to simply bury Howard. We must also
frameworks for expressing the meaning of design components within a single reasoning issue or across several reasoning issues [Hartmann 1990] [Bjork 1988] [Laiten 1994] [Amor 1993]. Product modeling research, such as the STEP efforts, has typically employed an essentially component-oriented approach.

The drawback of the component paradigm is that the comprehensive definition of component semantics is an extremely difficult task. It requires that the researchers or software developers preparing the object hierarchies must be fluent in many diverse design disciplines and must define terminology for design attributes that can be accepted across disciplines. There is a tendency toward a profusion of attributes on each object in order to accommodate any conceivable use of the object in a design project. Due to designers' well-known predilection to clever or even perverse use of design objects, one might speculate that real designers would take pleasure in breaking whatever system of component semantics that developers could devise.

Modular Design Environments

Both the layer-based approach and component-oriented approach provide predefined, rather inflexible semantic representations. At some point before designers can begin sketching and portraying design ideas, the semantic content of the project must be prescribed, either in a layer structure or in the attributes of a component hierarchy. Some researchers have recognized the limitations inherent in predefined design semantics. An emerging vision of a building modeling system that can be dynamically composed for a particular project [Eastman 1992]. An engineering environment could consist of application modules that communicate via messaging using an object-oriented operating system [Weisemann 1988]. An engineer or architect could collect preferred design software tools into a custom computerized workspace. A similar vision has led us to our concept of a virtual product model.

Figure 7 illustrates the difference between the component approach and the virtual product model approach. The component approach centralizes a large and sophisticated representation of building form to which evaluation tools are tightly connected. A virtual product model uses a minimal central representation of building form to which explicit representations of function and behavior may be dynamically linked. The evaluation tools use the representations of form, function and behavior for reasoning about a particular issue. The virtual product model idea decouples the graphic representation of form from the symbolic representation of function and behavior, allowing the designer to consider the drawing and the semantics separately and creatively. The product model in the SME prototype is intangible, rapidly reconfigurable, and adaptive to needs that arise during the course of the project. In these senses, it is a virtual product model.

![Figure 7: A component-based product model and a virtual product model.](image)

Implementation

Most of the development work for SME was done on Sun Sparc workstations, although versions of the software also run on DOS, Windows, and Macintosh platforms with reduced functionality. This section describes the software components and the development tools used for its implementation. Initial experience with SME has lead us to an understanding of tradeoffs resulting from the basic concepts and the particular implementation.
Software Components

The SME prototype consists of a CAD graphic system, an integrated virtual product modeling system, and several design evaluation tools implemented as interpretations. The CAD graphic system provides an interface for creation of form objects using constructive solid geometry. The virtual product modeling system implements an interpretation manager and interpretation base classes, establishing an interface for dynamically loading interpretations into the environment. It also provides an interface consisting of a few conceptually simple object methods that newly developed interpretations must support. Four evaluation tools implement form, function and behavior objects relevant to the issues discussed in the scenario.

Tools

We used AutoCAD and the AME solid modeler as the graphics editor for SME. The full array of AutoCAD’s construction tools are available to the designer. We felt that the constructive solid geometry tools of AME provided an intuitive and powerful 3D drawing environment. AME provides rectangular boxes, cylinders, spheres, wedges, extrusions and solids of revolution as well as Boolean operations of union, subtraction and intersection on any solid. The virtual product modeling system, the Semantic Modeling Extension proper, was implemented with Autodesk ADS C routines and AutoLISP functions. Autodesk development tools were used extensively for preparing user interfaces. The evaluation processes for each interpretation were written using the ProKappa object-oriented and rule-based development environment. SME provides interprocess communication services to send messages to each of the evaluation processes and receive messages back in the AutoCAD model.

Room for Improvement

As implemented, SME only approximates the concept of a virtual product model. For ease of implementation, most of SME was implemented using LISP along a functional programming paradigm and is not truly object-oriented. The interprocess communication technique for connecting each interpretation to a reasoning process works for demonstration purposes and has allowed for rapid prototyping, but at the cost of execution speed and conceptual purity. A full message-passing, object-oriented implementation has been planned [Clayton 1994]. A new implementation would provide improved encapsulation and simplified extensibility, as well as expected enhancements to execution speed. Routines for extracting geometric quantities from the CAD model are the slowest part of the system. However, they were written to be extremely general to any shape that can be defined with constructive solid geometry. Thus, the system places the smallest possible constraints upon the definition of form to permit testing in realistic design scenarios. We anticipate that advances in hardware and graphic modeling software will effectively reduce the execution speed of the extraction routines.

The evaluation modules are at present crude and simplistic. To test the ideas and system architecture conclusively will require a large effort to add knowledge to the software.

Implications and Conclusions

The design, implementation, and testing of SME has been undertaken within the context of speculations regarding improvement of the design process through computer-based technical innovation. This research has produced a testbed by which the usability and effectiveness of the concepts of a virtual product model may be explored.

Usability

Interpretations provide an integration of graphic and symbolic design representations that is distinctive and attractive. By allowing designers to draw their design ideas first and then apply the semantics necessary for evaluative reasoning, the SME prototype may appeal to a different segment of the design community than other computer-integrated design systems.

An open question that arose quickly in this project is whether an interactive approach to applying confront the mythologies that persist: the focus on creativity without context; the search for immortality in personal physical monuments; and the inflated expectations of “talent,” among others. This requires that we recognize the human motivations these myths imply and incorporate them in new mythologies or visions of the architect and architecture. In her illuminating book, "Architecture:
semantic values to graphic entities can provide benefits competitive with those promised by the predefined semantics of other integrated CAD research. Although the interpretation of designs using the tools incorporated into SME appears tedious, this activity can generally be performed rapidly and easily. Most of the interpretation activity is performed at the beginning of the development of a graphic model. As the design is adjusted in the course of design iterations, many changes to the graphic model do not require any changes to the interpretation and thus are updated automatically to the symbolic models.

Nevertheless, the use of interpretations appears most appropriate for early design stages in which the problem conditions and design solutions are not yet well defined. This approach allows the designer to withhold commitments and defer evaluation while using graphic tools to explore design ideas. The flexibility achieved could be very important during preliminary design, especially in non-routine problem-solving situations.

As an architectural design project progresses into detailed design, design decisions become more oriented toward selection from catalogues of components with well-defined functional characteristics. Perhaps the ideal integrated design system would allow the user to transition seamlessly from a virtual product modeling system like SME to a component-based product modeling system.

Circle Integration

One of the initiating speculations for this research has been the concept of circle integration [Fischer 1993]. The premise is that members of a design team may more rapidly achieve a consensus by employing design software that integrates some of the reasoning of each team member. Each team member may use the software to evaluate design ideas as if the whole team were present, “going around the circle” with each design change. Circle integration presumes a common design representation and automation of some of the reasoning provided by each member of the design team. It depends upon the provision of a means to dynamically edit the members of the circle to incorporate new viewpoints and delete viewpoints that are no longer relevant.

The SME prototype demonstrates that circle integration can be implemented using a commercial CAD system and a virtual product model.

Effectiveness

The research to date provides evidence that circle integration using a virtual product model, by virtue of automated design evaluation in response to multiple issues, may provide value in the architectural design process. Even with the simple prototype design criteria used in this research, interesting consequences and interactions resulting from design decisions emerge rapidly and are substantiated by accurate and concrete evidence. The execution of the design evaluation software, generally within seconds or minutes, is at sufficient speed that it does not overly interrupt the design thought process. The designer is equipped with much improved evaluation of design alternatives, allowing for more rapid design iterations, consideration of more alternatives, and potentially higher quality resulting designs.

Desktop Engineering

Analogous to desktop publishing, the concept of desktop engineering postulates that the capabilities of a design team may be replicated by a software tool used by a single designer [Kuz 1994]. Although this speculation is threatening to many professionals, it promises enticing benefits to building owners and significant increases in a design firm’s productivity. The reasoning provided by the SME prototype falls far short of the expertise necessary for a commercial desktop engineering system, but SME demonstrates the concept of desktop engineering and provides some evidence that commercial desktop engineering systems are achievable. An effective desktop engineering system could lead to the concentration of knowledge and authority for a building project under an individual who would more closely approximate the master builder, or architect, of past eras.

AutoCAD and AME are trademarks of Autodesk, Inc. ProKappa is a trademark of Intellicorp, Inc.

Howard's End

ACADIA 1994
References


The Story of Practice, Dana Cuff offers the vision of practice as a social process that shifts emphasis from individual creativity to collective action; from independent decisions to shared understanding; and from the contrasts between architects as artists-versus-businessmen (or generalists-versus-specialists) to the potentials of their interaction.


