CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

# Rapid Conceptual Design Evaluation Using a Virtual Product Model

By

Mark J. Clayton
John C. Kunz
Martin A. Fischer

**CIFE Technical Report #105**
**December, 1995**

**STANFORD UNIVERSITY**

# SUMMARY
## CIFE TECHNICAL REPORT # 105

**Header:** SME

**Title:** *Rapid Conceptual Design Evaluation using a Virtual Product Model*

**Authors:** Mark J. Clayton, John C. Kunz, Martin A. Fischer

**Publication Date:** 12/95

**Funding Sources:**
Name of Agency: CIFE
Title of Research Project: "Circle Integration and Desktop Engineering"

**Abstract:** This paper presents an architecture and prototype test results of a computer-based system for assisting the conceptual phase of building design. The system uses 3D CAD to represent a graphic model of the facility design, and it builds a symbolic model of the form, function and behavior of the design. The system includes automated methods to analyze design behavior and compare predicted behavior with intended function. A virtual product model-based CAD system was implemented using the concept of interpretations. After drawing a design using 3D CAD, a designer interactively creates interpretation objects to express the meaning of the graphic representation with respect to a particular issue. The interpretation represents geometric and topological attributes of the features for use by automated design analysis tools. Interpretation objects unite support for graphically-oriented design thinking with support for automated symbolic reasoning. The paper includes an example building design scenario using the software prototype, illustrating how interpretation of the geometric model produces a symbolic model and supports multiple and changing analyses and evaluations during design. The system implements Circle Integration, and it is an example of Desktop Engineering. The system has been tested by students and practicing engineers.

## 2. Subject:

- What is the report about in laymen's terms? CAD models of buildings normally are purely graphic: they cannot describe the intended functions or most of the predicted behaviors of the design. This system shows a way to develop such design data concurrently with development of the CAD model design itself.
- What are the key ideas or concepts investigated? behavior, CAD, charette, circle integration, evaluation, form, function, interpretation, Virtual product model,
- What is the essential message? "interpretation" of the geometric CAD model produces a symbolic model and supports multiple and changing analyses and evaluations during design.

## 3. Objectives/Benefits:

- Why did CIFE fund this research? We proposed to develop and implement a theory of software applications integration that would support multiple, rapid and changing analyses and evaluations during design.
- What benefits does the research have to CIFE members? The results of our charette testing to date provides evidence that circle integration using a virtual product model,

**(Summary of TR #105 continued)**

by virtue of automated design evaluation in response to multiple issues, can provide significant value in the facility design process. Even the prototype system provides the designer with much improved evaluation of design alternatives, allowing for more rapid design stages, consideration of more alternatives, leading to designs of potentially higher quality.

- What is the motivation for pursuing the research? A great deal of research has been done to support automation and integration of existing design and analysis processes. We were motivated to propose ways to reengineer the design process to provide significantly better design information far more rapidly than is possible using methods that depend on the slow conduct of social design interactions.
- What did the research attempt to prove/disprove or explore? This project formulated and tested the hypothesis that SME use provided no change in design quality in comparison with manual methods practiced by either students or practitioners. The results strongly deny the equality hypothesis: both students and industry practitioners developed more and better design alternatives using SME than using manual methods.

## 4. Methodology:

How was the research conducted? We used several development methods:

- Represent graphic forms of a design and, separately, a symbolic design model.
- Automate prediction of behavior.
- Automate comparison of behavior with respect to function.
- Implement principled knowledge representation.
- Implement circle integration architecture.
- Test the SME system with a real test case using both students and professional engineers.

Did the investigation involve case studies, computer models, or some other method? Modeling of industrial case study.

## 5. Results:

- What are the major findings of the investigation? Both students and industry practitioners developed more and better design alternatives using SME than using manual methods.
- What outputs were generated (software, other reports, video, other) This report.

## 6. Research Status:

- What is the status of the research? Completed
- What is the logical next step? Automate interpretation, which is now manual, and move the technology to the web.
- Are the results ready to be applied or do they need further development? Both
- What additional efforts are required before this research could be applied? Basic ideas are now useful. Tools and methods need development.

# Rapid Conceptual Design Evaluation using a Virtual Product Model

Mark J. Clayton
Department of Architecture
Texas A&M University
College Station, TX 77843-3581 USA

John C. Kunz[1]
Center for Integrated Facility Engineering
Stanford University
Stanford CA 94305-4020 USA

Martin A. Fischer
Department of Civil Engineering
Stanford University
Stanford CA 94305-4020 USA

## Abstract

This paper presents an architecture and prototype test results of a computer-based system for assisting the conceptual phase of building design. The system uses 3D CAD to represent a graphic model of the facility design, and it builds a symbolic model of the form, function and behavior of the design. The system includes automated methods to analyze design behavior and compare predicted behavior with intended function. A virtual product model-based CAD system was implemented using the concept of interpretations. After drawing a design using 3D CAD, a designer interactively creates interpretation objects to express the meaning of the graphic representation with respect to a particular issue. The interpretation represents geometric and topological attributes of the features for use by automated design analysis tools. Interpretation objects unite support for graphically-oriented design thinking with support for automated symbolic reasoning. The paper includes an example building design scenario using the software prototype, illustrating how interpretation of the geometric model produces a symbolic model and supports multiple and changing analyses and evaluations during design. The system has been tested by students and practicing engineers.

## Introduction

The Semantic Modeling Extension (SME) system allows the user to:

- **Create conceptual designs using tools that are natural.** With SME, the designer uses 3D CAD.
- **Develop and evaluate a conceptual design significantly faster than using traditional methods**. Evaluating a design now takes days or weeks of calendar time, although not necessarily more than a few days of actual work time. The objective is to use automation to analyze design behavior and to evaluate the extent to which behavior conforms to functional requirements.
- **Generate and evaluate many design versions.** Designers are perfectly able to generate design versions; time pressures constrain them. By engineering the design process to support automation and integration of prediction and evaluation tools, we want to support very rapid (minutes vs. days) design behavior prediction and evaluation and enable rapid creation of design versions.

[1] To whom correspondence should be sent.

- **Extend and modify the facility product model easily.** We assume that no predefined library of product models will satisfy the continuing needs and preferences of diverse projects and designers. Our objective is to support facility models that use a generic set of *foundation classes* that users can extend as needed.

To achieve these goals, we used several development methods:

- **Represent graphic forms of a design and, separately, a symbolic design model.** We represent the graphic forms in 3D CAD. We represent the symbolic description of forms, functions and behaviors of the design in an object-oriented knowledge representation and manipulation system. The graphic representation supports graphic design and graphic communication of the design. The symbolic representation supports behavior prediction, evaluation and explanation.
- **Automate prediction of behavior.** The current prototype uses a set of integrated prediction methods to identify geometric properties of a design such as volumes and distances, architectural properties such as egress paths, construction properties such as cost, and functional properties such as solar incidence and energy loss.
- **Automate comparison of behavior with respect to function.** The system uses an integrated set of evaluation methods to identify discrepancies between designed behavior and function, e.g., conflict between cost and budget, between designed and required space volumes, between planned and good-practice solar incidence.
- **Implement principled knowledge representation.** For each geometric feature of a design, SME represents its geometric forms, functional design intent, and predicted design behaviors. The symbolic forms, functions and behaviors constitute the foundation classes in the SME product model.
- **Implement circle integration architecture**. The SME analysis software modules are linked together in a "circle" integration architecture [Fischer]. This architecture integrates prediction and evaluation processes with a control structure that enables simple and predictable system behavior.
- **Test the SME system with a real test case using both students and professional engineers.** The *charette* test method provides feedback to SME developers about the effectiveness of each SME version.

The SME architecture places the specification of function and act of drawing first in the design process. After drawing a design idea using a computer graphic system, the designer identifies the semantic content of graphic forms by annotating each graphic entity with its intended functions. An interpretation expresses the meaning of the design within a particular context, such as structural sufficiency, energy consumption, or requirements for egress. A design may have many interpretations to express complex and multidisciplinary semantics. As the design model now represents both the geometry and the semantics, knowledge-based and algorithmic critiquing tools may be used to evaluate the sufficiency of the design behavior with respect to the design functional intent. In a new design version, the designer changes the form or function of the design to resolve conflicts.

## Background

Numerous research prototypes of integrated CAD systems have been built. Typically, these systems manipulate a database, use rules, apply constraints, or in general use some computer science approach to synthesize graphic forms from a statement of function. Typically, these approaches tightly bind semantic values to the graphic representation. The design process becomes a process of component selection. There remains an uneasiness that such computer synthesis methods diminish the creativity and delight inherent in traditional drawing-based design methods.

SME distinguishes drawing a design from assigning meaning to a graphic form. The SME user *interprets* graphic forms to identify their symbolic form, functions and behaviors. The explicit interpretation approach used by SME builds on the rich and well-established tradition of graphic thinking for design. The designer to interactively interprets the semantics of the graphic representations created during design explorations.

The concept of a virtual product model, described in detail below, provides a foundation for the creation of computer-aided design (CAD) systems that allow designers to work in a relatively free-form graphic environment yet obtain many benefits of automated design evaluation and prediction software. A virtual product model explicitly and separately represents the design form, function and behavior. The designer can independently manipulate the form and its interpreted functions. A virtual product model avoids the necessity of a rigidly defined construction component library, yet provides the designer with the benefits of automated critiques integrated to a graphical CAD system. By providing easy-to-use tools for linking graphic representations of the building form to symbolic representations of function and behavior, a virtual product model achieves an extensible, flexible environment for the early, exploratory stage of design.

## *Scenario*

An architect needs to design a suite of rooms for a hospital. This scenario is built on observations of a recently-completed building project. The architect has a top-level objective: design spaces to accommodate equipment (a cyclotron) that will provide radioisotopes for a radiology lab. The architect's role is to specify the spaces, enclosures, and equipment in enough detail to permit construction, while meeting budgetary and functional requirements and aesthetic goals.

The hospital administrator has provided the architect a 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 administrator also provides contacts at the cyclotron manufacturer and brochures describing the equipment.

### Initial drawings

The architect's initial step in producing a design is to formulate the design, as a conceptual-level 3D CAD model and as a statement of building functions. Architects call the statement of function the "architectural program." The architect lays out a basic volume to accommodate the cyclotron and the radiology lab and then sketches 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 sketch. As the drawing is 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.

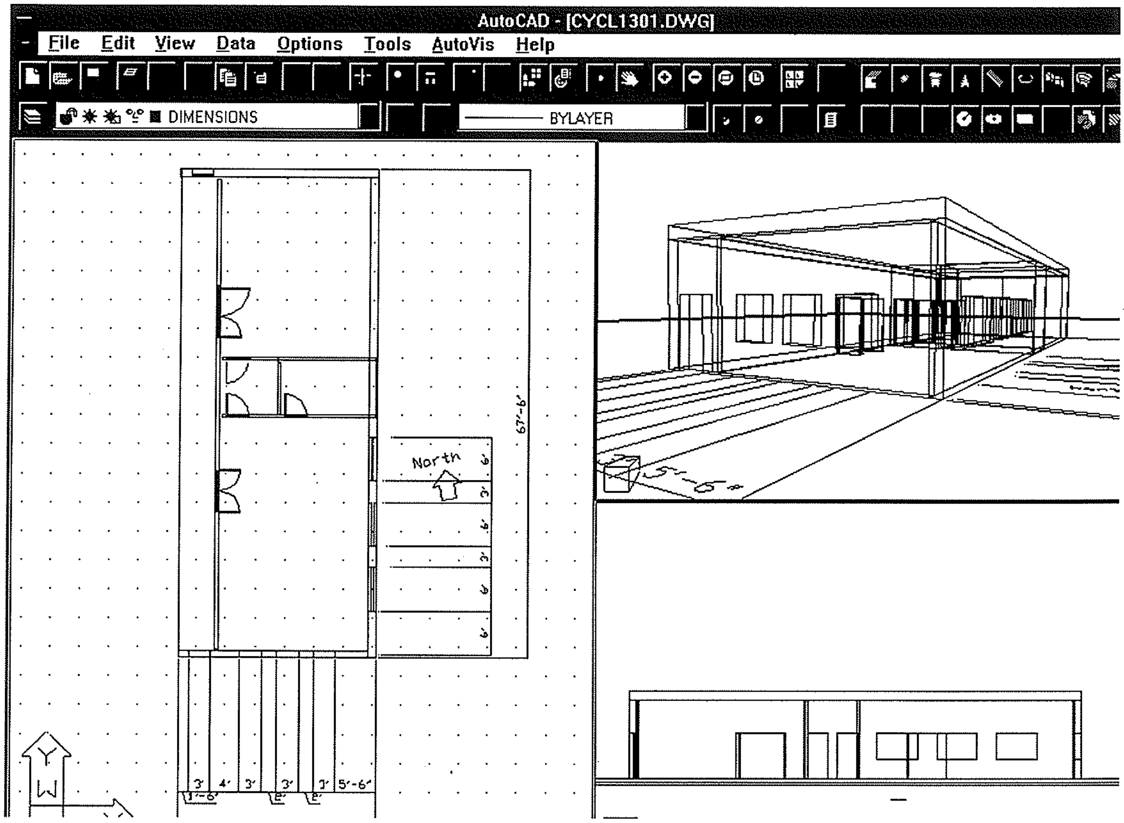
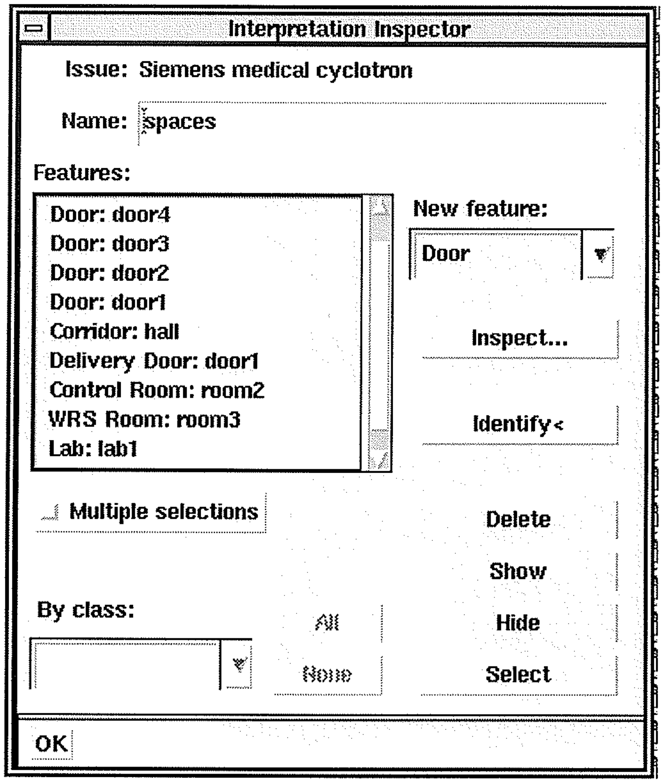


**Figure 0-1: Initial sketch of the cyclotron room for a hospital. The SME user creates a design using 3D CAD. The user then interprets the components of the graphic model to build a symbolic model. SME analyzes the symbolic model to predict behaviors of the designed components, and it evaluates consistency of behavior and design functional objectives.**

## Interpretation of graphic forms

The designer creates graphic entities in the CAD model. At this point, the solids drawn to represent the elements in the design have no explicitly expressed semantics, and thus automated evaluation is not possible. In the process of *interpretation,* the designer then creates symbolic entities to describe each form and links corresponding graphic and symbolic form entities. The symbolic form objects reference symbolic functions that describe the design intent of including the forms. Clients specify some functions; code and industry practice determine others; needs and preferences of contractors, suppliers, operators and users specify others. Finally, forms have associated predicted behaviors.



Issue: Siemens medical cyclotron

Name: spaces

Features:

- Door: door4
- Door: door3
- Door: door2
- Door: door1
- Corridor: hall
- Delivery Door: door1
- Control Room: room2
- WRS Room: room3
- Lab: lab1

New feature: Door

**Figure 2: Adding an Interpretation to the CAD system. Interpretation functions allow the designer to annotate graphic form entities with functional intent. The designer interactively annotates relevant graphic form entities with semantic interpretations. The symbolic interpretations get expanded into the symbolic form, function and behavior objects of the facility model.**

Using the `Interpretation Inspector` dialog box shown in Figure 2, the architect interactively annotates relevant graphic form entities in the CAD model with semantic interpretations. By clicking on the New button, the architect can designate which class of feature to create, the feature name, and what graphic entity to associate with this feature. The architect elaborates a suite of rooms by drawing volumes to represent the individual rooms. He identifies each room in the spaces interpretation, such as the cyclotron room, the radiology lab, and the water recirculating 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 editing the feature class or its attribute values. At any time, the user can associate a graphic entity to a different feature. For instance, a storeroom may initially be designated as a custodial store but then, because of proximity to the cyclotron room, be redefined as a mechanical space for the cyclotron support equipment. The designer can control the graphic display of the design by hiding, showing, or highlighting the graphic entities that correspond to particular features.

## Egress, cost and energy interpretations

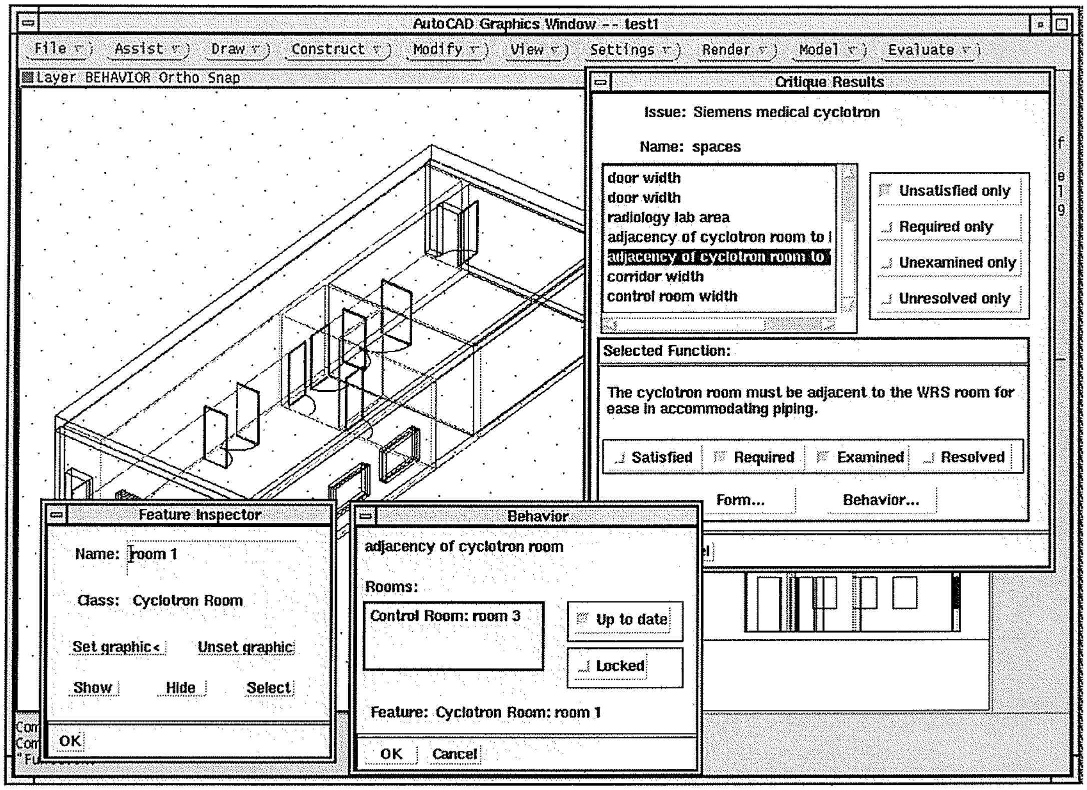
To check that the building must satisfy code requirements for egress, the architect loads an egress interpretation. The hospital administrator wants preliminary cost estimates for the new wing and a construction schedule. Using the Interpretation Manager, the architect installs a construction cost interpretation into the CAD system. The architect sketches in the footings, HVAC units, lighting systems and equipment so that the physical components of the building are all shown as solid objects. During preliminary design, the components are drawn in a fairly abstract representation. Using the Interpretation Editor, the architect designates the meaning of each drawn object within the context of the construction interpretation by identifying construction features. The construction features refer to items in an assembly-based estimating database such as spread footings, block walls, and drywall partitions [Means]. The architect also types in values for the cost budget and the project time constraints.

To address the energy consumption issue, the architect loads another interpretation for examining the energy consumption of a building. Once again, he 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. The architect designates the climate for the project and a set of energy consumption objectives.

## Design critique

Having annotated graphic forms with their functional intents, the architect now invokes an automated prediction of the behavior of the design and critiques its behaviors with respect to functions. In a few seconds, a reasoning module analyzes the design and evaluates its sufficiency 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 provides the evaluation that the design meets the relevant code provisions for egress by examining the size and connectivity of rooms and doors in the egress interpretation. A construction evaluation tool provides a cost estimate and a preliminary construction schedule. In a few more seconds, another tool evaluates the design regarding its predicted consumption of energy.

Each automated critic provides an explanation of its results and suggestions for improvements. This information has links to the 3D CAD model so the user can display a textual explanation and highlight affected objects. The architect can view the critique results within the context of the actual design. The `Spaces` critique suggests changes to the spaces and doors to accommodate better the installation of the cyclotron. The egress critique is that the building satisfies the code requirements. The construction critique annotates the design objects with component costs and initial construction dates. Figure 3 illustrates the results generated by the spaces critique that a room needs to be adjacent to another room. The `Feature Inspector` in the lower left window of Figure 3 shows "room 1" as the cyclotron room. The `Behavior` window shows that this room is adjacent to "room 3," a control room. The `Critique Results` window in the upper right shows that this room should also be adjacent to another room to resolve a particular engineering issue.



Issue: Siemens medical cyclotron

Name: spaces

door width
door width
radiology lab area
adjacency of cyclotron room to I
adjacency of cyclotron room to
corridor width
control room width

Selected Function:

The cyclotron room must be adjacent to the WRS room for ease in accommodating piping.

Name: room 1

Class: Cyclotron Room

adjacency of cyclotron room

Rooms:

Control Room: room 3

Feature: Cyclotron Room: room 1

**Figure 0-3: Results of the `Spaces` Critique. The system analyzes behavior of the design, compares predicted performance with design intent, and suggests appropriate design modifications to meet functional objectives that are not being satisfied.**

The prediction and evaluation process leads to a set of suggestions about design conflicts. The architect chooses one or more conflicts to resolve, changes the form appropriately, and then invokes the prediction and critique 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 graphic form may include drawing new objects, changing old objects, and re-interpreting the semantic functions of graphic entities. Most of the work is in editing the form of the building using the graphic editing tools.

## Presentation

The interpretation and initial prediction and evaluation process takes 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 he can show to the client. Each of the descriptions is in the form of a critique result expressed as an annotation of the 3D graphic model.

The architect and hospital administrator examine the issues together. By hiding the graphic model related to some interpretations and showing the graphics related to others, the architect and administrator can focus upon a single interpretation or combinations of interpretations. 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 discussing 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 instructions from the client. He obtains another critiquing tool from the second cyclotron manufacturer, perhaps down-loading it across a wide-area network, and loads it into the SME 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 architect uses automated critiques of the design to explore alternatives and balance the various issues. By carefully adjusting the dimensions of the cyclotron room and lab space, the architect achieves a design that satisfies both cyclotron alternatives. He explores the cost and energy impact of changing the exterior walls to brick construction and adding the skylights to the lab space by examining the results of the construction and energy evaluation tools. The critiques of each alternative provide a documented trail of the evolution of the design and guidance toward improving performance.

At the end of the day, the architect has studied several design alternatives and 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. It partially automates the design prediction and evaluation. It considers multiple issues in design. It provides mechanisms for accommodating changing information needs as a project progresses. It integrates graphic thinking and symbolic analysis, allowing for free consideration of the design in both modes.

Design involves an iterative cycle of activities in which a design idea is formulated in response to requirements and tested for sufficiency to those requirements. [Asimow] describes these activities 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. SME assumes a slightly refined description of the design process. Figure 4, based on [Clayton] and [Gero 90], shows a design interpretation node in the SME system for one issue of one engineering discipline. Each design interpretation node supports a particular functional issue, such as cost estimation, energy use or building egress. The user and SME together perform the steps shown in Figure 4:

- step-0: the SME user describes intended functions of the facility being designed and creates a project description. Normally, the project description includes a CAD model, the architectural program, and other documentation such as a schedule and references to parts in a component pricing catalog. In the SME system, the CAD model is created using a small set of 3D CAD primitives defined in the AutoCAD AME system.

- step-1: the user manually "interprets" the project description, or identifies 3D objects in the CAD model as symbolic forms (e.g., doors, corridors) that are relevant to the particular issue. This interpretation process creates a symbolic model of the facility design.

- step-2: SME methods automatically compute the predicted behavior of form entities.
- step-3: SME methods, called "critics," automatically compare the consistency of predicted behavior with intended function. SME critics now compute and check construction cost, facility egress, energy use, and architectural space use.
- step 4: Following an evaluation, the user can manually change symbolic description of function or form to resolve any reported conflicts.
- step5: Following any reformulation of the symbolic model, the designer manually updates the project description and the CAD model.

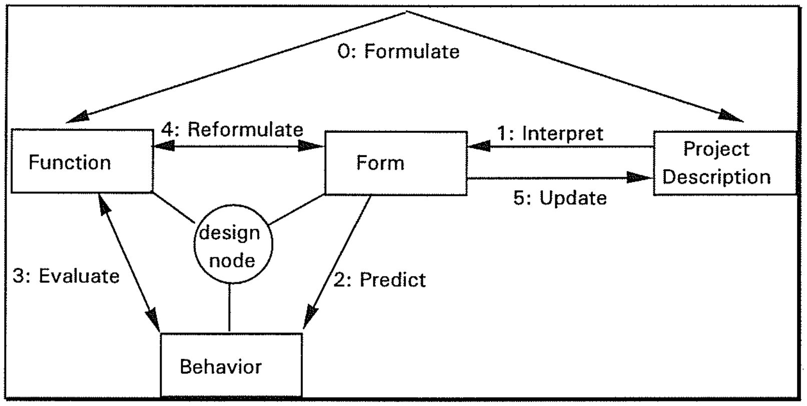


**Figure 4: Design interpretation node in the SME system: An interpretation node links specific Function, Form and Behavior foundation class objects for a specific design objective with a general project description. The SME system implements automated prediction and evaluation methods (steps 2, 3) for several engineering disciplines. The SME user manually now performs steps 0, 1, 4 and 5. Interpretation and update may be readily amenable to amenable to partial automation.**

The scenario describes a process in which the designer may rapidly iterate through steps in the design cycle. SME automates the prediction and evaluation. They are often bottlenecks in the design cycle. First, they often are performed by specialist consultants after talking with an architect, and it often takes far longer to schedule design meetings than to perform the prediction and evaluation. In addition, prediction and evaluation require making judgmental and difficult abstractions, followed by tedious calculations that must be repeated for each alternative studied. One result of the difficulty of this social interaction is that there often is client resistance to the cost of considering multiple design options. SME attempts to relieve the prediction-evaluation bottleneck by automating simple, preliminary design prediction and evaluation. Integrated, automated critics can drastically reduce the time necessary for considering multiple design options during conceptual design, from a prohibitively long period of several days to an inconsequential few seconds. As a result, the process described in the scenario is significantly accelerated over non-computerized methods, allowing the designer to consider more alternative designs. The computerized prediction and evaluation tools also provide more rigorous, more consistent and higher quality documentation of the performance of alternatives than is generally prepared using manual or non-integrated methods.

The SME prototype does not now support analysis of many important issues, such as structures or constructability, nor are its analyses exhaustive or complete. It does not automate design synthesis to resolve conflicts between design behavior and function. Over time, the design analyses and evaluations will become more extensive and more complete. Different specialists such as architects and structural engineers will start to use tools like SME, and they will share their design versions across a computer network. We expect that, for designs in which aesthetics are important or in which there are inherent conflicts among competing objectives, the actual synthesis process will continue to be carried out solely in the mind of designers.

### Multiple issues

Design, especially at the early stages, involves the consideration of multiple issues and their integration into a balanced whole. To achieve a balance, architects alternate rapidly among contexts of reasoning, as revealed by protocols of designers at work [Schon]. 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]. 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. By its list of available classes, each interpretation establishes a vocabulary of features that expresses the semantic content of the design within a particular context.

Examination of interactions and dependencies across the various interpretation contexts is supported by providing Boolean operations upon sets of interpretations. The lists of graphic entities referenced by an interpretation may be unioned, subtracted, or intersected with the entities referenced by another interpretation and the resulting list of entities hidden, displayed, or highlighted on the graphics window. The architect may also query a graphic entity to discover which interpretations reference the entity.

### Changing issues

At the early stages of a project, the designer must identify the engineering issues that affect design quality. As the owner juggles budgets and schedules, he may change the objectives for the project or defer some objectives to a later project. The architect is on the lookout both for emerging opportunities and the client's unstated objectives. In the scenario, the hospital administrator adjusts the functional requirements of the building addition by requesting that it accommodate cyclotrons 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. The informational structure for such tools must be adaptable to new information needs and relations. This capability has been explored in design database research in the GLIDE project and has been referred to as "dynamic schema extension" [Eastman]. To accommodate this need for changes to informational needs, the SME prototype provides for dynamic loading of interpretations 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 interpretation design system and associate features with particular graphic entities in the CAD model.

## Graphic thinking and symbolic analysis

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

Computational methods of reasoning, however, depend upon symbolic analysis. 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 [Milne]. 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] [Pohl].

The SME prototype integrates tools for both graphic thinking and symbolic analysis. In the scenario, the architect first uses graphic thinking to compose a design synthesis. Evaluative thinking is suspended 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 analysis 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.

## Charette testing

We tested the effectiveness of the SME system using a series of design "charettes." Our charettes were short (60-90 minute) design evaluation tasks. We used both students and professional engineers as testers. Each tester received a three page design program that describes the function of the different rooms in the cyclotron suite of a local hospital, as described in the Scenario section above. The program also described certain special requirements such as budget and the need of the cyclotron room for radiation shielding. In addition, each tester was given a graphic design. Testers were divided into two groups: manual testers and SME users. The manual testers received a paper drawing of the design; the SME users were shown the 3D facility model as shown in Figures 1 and 3. Finally, each manual tester was given a four page database of component prices. The SME user had the contents of the database in the computer, but the values were slightly different so that some of the analysis results would be different. The charette method was to ask each tester to analyze the design two times, once manually and once using the SME system. Half of the testers did the manual process first, then used SME; the other half of the testers used SME first. The purpose of the controlled randomized trial was to allow us to control for both the design analysis skill of testers and the learning effects of the first trial. We asked each tester to evaluate the design with respect to conformance to requirements on cost, egress, providing required spaces, and energy use. In addition, we asked testers to propose design modifications to the initial design they received – a partial design that deliberately included a number of problems.

We measured several aspects of tester performance, including:

- Problems properly identified;
- Number of design versions and time to produce each; and
- Quality of the modified design with respect to the carefully considered final design of one of us (Clayton).

The results of the design charette showed unequivocally that SME users were more effective than users of manual analysis:

- SME users properly identified significantly more problems than users of the manual procedure;
- SME users typically produced 1-3 new design versions; most manual users did not complete a single redesign;
- The quality of the modified design proposed by SME users was generally improved over their initial partial design.

## *Architecture of the SME Virtual Product Model*

The SME prototype is based upon a virtual product model paradigm using the concepts of form, function and behavior. The STEP model, for example, explicitly represents form in great detail, but it does not explicitly represent function and behavior [ISO]. 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. In support of conceptual design, the virtual product model defines just enough detail about form to allow associated representation of function and prediction of behavior.

### 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 references associated function and behavior. 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 design 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 Formulate step in Figure 4 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 radioisotopes. It leads to requirements about minimum sizes of doors, access between rooms, adjacency of one room to another, and dimensional requirements for spaces. However, 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. Although the prototype now represents cost simply as a single figure, it could be represented in a more complex and sophisticated manner such as a range, or a breakdown into cost categories. 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 to 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 design situation, concerning a particular engineering issue. Behavior is typically derived from the form model by a process of prediction [Carrara]. Behaviors normally 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 others, e.g., [Luth].

Behaviors are measurable or observable attributes of design forms, such as the examples shown below in Figure 5. Many authors limit behavior to those attributes that are computed (e.g., volume, cost), assigning to form those geometric attributes that are assigned (e.g., length, width). Although the definition is arbitrary, we chose to group both assigned and predicted values under the generic class of behaviors. This convention leads to a very simple definition of evaluation: it compares behaviors with functions.

Several behaviors have been defined. 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.

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 wall 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.

Figure 5 shows examples for symbolic and graphic forms and symbolic functions and behaviors. Each design issue has different forms, functions and behaviors.

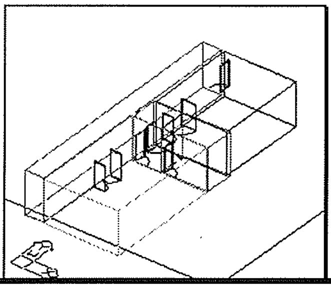
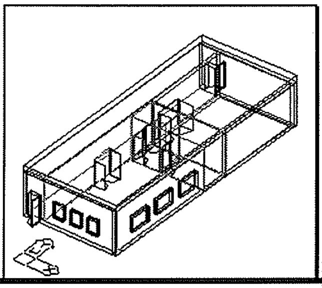
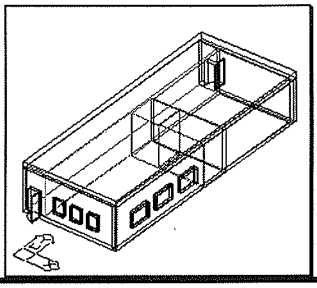
| Issues: | Space | Cost | Energy |
|---|---|---|---|
| **Forms:** | Geometry and materials<br> | Geometry and materials<br> | Geometry and materials<br> |
| **Functions:** | required rooms<br>required dimensions<br>required floor area<br>required room adjacency<br>required room connectivity<br>required door size<br>required wall construction | project budget<br>budget per category | heating budget<br>cooling budget<br>energy budget per category |
| **Behaviors:** | existence of rooms<br>length<br>width<br>height<br>area<br>room adjacency<br>room connectivity<br>door size | length<br>width<br>height<br>area<br>volume<br>total cost<br>cost by category<br>item cost | surface area<br>volume<br>orientation<br>adjacency of spaces, walls<br>connectivity of windows, walls<br>energy flow through each item<br>temperatures |

**Figure 5: Examples of symbolic descriptions of forms, functions and behaviors. The designer annotates the graphic CAD model with symbolic descriptions of forms and functions. The SME system predicts behavior.**

## Interpretations

In an individual product model, symbolic form, function and behavior are all represented as distinct objects. The aggregate of a particular form and its associated functions and behaviors is called an *interpretation* for an issue of a design. SME methods predict behavior and evaluate function. They work with respect to an interpretation. Technically, a *feature* object is created for each interpretation to hold the pointers that link individual forms with their associated functions and behaviors.

Each design interpretation node contains a Function-Form-Behavior triple and a reference to a project description. Thus, an interpretation node implements each design objective considered by the system. Each interpretation node also has methods to insert itself into an integrated suite of interpretation nodes and to pass control to successor nodes.

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 the form, functions and behaviors that are relevant to the reasoning of an issue. The Egress interpretation, for example, defines Doorways that connect two spaces, Entrances that connect a space to the outside, and several kinds of spaces, including Classrooms, Storerooms, Corridors. The Construction interpretation defines an extensive list of construction assemblies derived from a published industry cost reference, e.g., the Means assembly cost data [Means]. The Energy interpretation defines Walls, Doors, Windows, Roofs, Slabs, and Overhangs, Fins, and other building features that impact energy flows and energy consumption.

### Interpretation Manager

Support for dynamic schema extension is achieved using an interpretation manager. This object maintains the list of loaded interpretations and allows the end-user to add new interpretations and to delete interpretations that are no longer needed. The interpretation manager also supports functionality that ranges across interpretations. Implemented in the prototype are capabilities for Boolean operations on the feature lists of interpretations and query of a graphic entity for its membership among multiple interpretations. Refinements to the interpretation manager could implement consistency checks among non-graphic attributes of features across interpretations.

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. By using the interpretation paradigm, automated design evaluation tools could be connected to any CAD system that provides an interpretation interface.

## *Related Research*

The evolution of computer graphics into true computer-aided *design* systems depends upon resolution of the puzzle of what are the relevant design semantics and how to represent them. In the commercial world, the approach has generally been to encode meaning into drawings by use of drawing layers. Researchers have generally adopted a component-oriented approach. Virtual product modeling presents a third method that has some interesting characteristics.

### Layer paradigm

Virtually all commercial CAD systems provide the functionality 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 to collect entities with related meanings, often by professional discipline, such as all structural elements onto one layer, all architectural elements onto another, all mechanical elements onto a third layer.

Unfortunately, the layer paradigm has several critical limitations. A layer name can only provide a semantic identifier to a collection of entities. The semantic content of individual entities on the layer cannot be represented explicitly in the layering scheme. The use of complex primitives, known variously as blocks, cells, or symbols, provides a means for naming individual entities. A more serious limitation is that an entity may exist on only one layer. Building designs can rarely be decomposed into collections of entities with single, distinct meanings. For example, a wall may appropriately concern architectural, structural and mechanical issues, but it can appear on only one layer. Many work-arounds have been devised for addressing this limitation, but none of them is fully satisfying. The layer paradigm, while appropriate to automating overlay drafting, is inappropriate to support detailed analysis of functions and behaviors required to evaluate designs of buildings. Nevertheless, layer-based representations of building semantics are widely used in commercial CAD tools, such as ASG and LightCAD.

### Component paradigm

The layer paradigm has been 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 purposes. The KAAD system, for example, provides components for the design of hospital suites that may be viewed at several degrees of abstraction and analyzed within several evaluative contexts [Carrara].

The component paradigm has been widely adopted in efforts in product modeling. Researchers devote attention 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 using relational databases, object-orientation, and frame-based semantic networks. Many interesting projects have proposed frameworks for expressing the meaning of design components within a single reasoning context or across several reasoning contexts [Bjork] [Luiten].

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 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 developers devise.

The component approach perhaps suffers from an over-emphasis upon the physical and non-physical objects that compose a building, without careful consideration of the purpose of those objects.

### Modularized 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 proscribed, either in a layer structure or in the attributes of a component representation. Some researchers have recognized the limitations inherent in predefined design semantics. An emerging vision is of a building modeling system that can be dynamically composed for the purpose of a particular project [Eastman]. An engineering environment could consist of application modules that communicate via messaging using an object-oriented operating system. An engineer could collect preferred documentation and evaluation tools into a custom computerized workspace. These objectives have led us to our concept of a virtual product model.

The virtual product model idea differs from the component-oriented paradigm because it creates relationships among explicitly represented semantic and graphic entities, rather than unifying the two. The semantics of each entity are considered only within a reasoning context that defines function and behavior representations. The semantics are defined interactively by the architect or engineer by identifying features within an interpretation. During the course of the building design effort, new entities may be added to the design, new semantics may be added to entities, and old semantics may be removed. The product model is intangible, rapidly reconfigurable, and adaptive to the needs that may arise.

## *Implementation*

The SME prototype implements a virtual product model for building design. Most of the development work 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, development tools, and class definitions used for its implementation. Initial experience with the implementation has led us to an understanding of tradeoffs resulting from the basic concepts and the particular implementation.

Details of the definitions for the base classes of Feature, Interpretation and Interpretation Manager are presented in [Clayton.]

### Components

The SME prototype consists of a CAD graphic system, an integrated implementation of a 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 publishes an interface consisting of a few conceptually simple object methods that newly developed interpretations must support.

### Tools

We used AutoCAD and the AME solid modeler as the graphics editor for SME. 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. Subtraction operations were used to define openings in walls and the roof as the most intuitive method of constructing doorways and windows. The full array of AutoCAD's construction tools are available to the designer. 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 Kappa object-oriented and rule-based development environment. SME uses interprocess communication to send messages to each of the evaluation processes and receive messages back in the AutoCAD model.

## *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 of the concepts of virtual product models and interpretations may be explored. It shows that technical integration of CAD and analysis functions is achievable. It suggests that this technical integration can be effective in increasing the number of design options considered and ultimately in improving design quality. It points to future systems that could significantly alter the composition and effectiveness of design teams through desktop engineering.

### Usability

Interpretations provide an integration of graphic and symbolic design representations that is distinctive and attractive. By allowing for 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 was whether an interactive approach to applying semantic values to graphic entities would 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, in actuality 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 response to evaluation, many design 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. The approach allows the designer to withhold commitments and defer analytical thinking 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.

## Tradeoffs

There are prominent tradeoffs resulting from the basic concepts behind SME. A virtual product model sacrifices some power and automation for increased flexibility. The interpretation process in SME requires manual activity, although some of it could probably be automated. As the evaluation tools execute in merely seconds, the interpretation process is potentially a bottleneck in completing a design iteration. However, the interactive approach to interpretation provides generality and flexibility to the system. The designer can analyze any design that can be mapped to an evaluation tool with a reasonable degree of engineering judgment. Component-based integrated CAD automates the interpretation process but at a sacrifice in flexibility, as the components provide only the information that has been pre-defined by the developer.

## Circle integration

One of the initiating speculations for this research has been the concept of circle integration [Fischer]. The premise is that a designer may more rapidly achieve a design that satisfies multidisciplinary design objectives by employing design analysis software that integrates reasoning of multiple disciplines. Circle integration can support both multi-participant collaborative design and desktop engineering in which a chief designer or a small design team develops the design with little support from specialist consultants. In collaborative design, 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 issue considered in the design. The SME prototype demonstrates that circle integration can be implemented using a commercial CAD system and interpretations that link graphic elements to a symbolic form-function-behavior product model.

## 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 [Kunz]. Although this speculation is threatening to many professionals, it promises enticing benefits to building owners and significant increases in a design firm's competitiveness. The reasoning provided by the SME prototype falls far short of the expertise necessary for a commercial desktop engineering system, but it demonstrates the concept of desktop engineering and provides some evidence that commercial desktop engineering systems are achievable. Desktop engineering could lead to the concentration of knowledge and authority for a building project under an individual who can be more responsive to building owners than a diverse team. Such an individual might act as a project director, spanning not only across disciplinary responsibilities but also across lifecycle stages of a building.

### Effectiveness

The results of our charette testing to date provides evidence that circle integration using a virtual product model, by virtue of automated design evaluation in response to multiple issues, can provide significant value in the architectural design process. Even with the simple prototype design critics 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 fractions of a second, is at sufficient speed that it does not significantly interrupt the design thought process. The designer is equipped with much improved evaluation of design alternatives, allowing for more rapid design stages, consideration of more alternatives, leading to designs of potentially higher quality.

## References

[Asimow] Asimow, Morris, *Introduction to Design*, Prentice-Hall, Englewood Cliffs, NJ, 1962.

[Bjork] Bjork, B-C, "A Unified Approach for Modeling Construction Information," *Building and Environment*: 27, 2, pp. 173-194, 1992.

[Fischer] Fischer, M. and Kunz, J. (1995). "The Circle: An Architecture for Software Integration." *Journal of Computing in Civil Engineering*, ASCE, 9(2), pp. 122-133.

[Carrara] Carrara G., Kalay, Y.E., and Novembri, G. (1994). "Knowledge-Based Computational Support for Architectural Design," in *Knowledge-based Computer-Aided Architectural Design* (Carrara & Kalay, eds.), pp. 147-201, Elsevier, Amsterdam.

[Clayton] Clayton, M.; Fischer, M.; Kunz, J.; Fruchter, R.; "Behavior Follows Form Follows Function: A theory of design Evaluation," *ASCE Congress on Computing in Civil Engineering*, 1995.

[Dixon] Dixon, John R., "Artificial Intelligence and Design: A Mechanical Engineering View," *AAAI-86: Proceedings, Fifth National Conference on Artificial Intelligence*, Morgan-Kaufmann Publishers, Inc., Los Altos, CA, 1986.

[Eastman] Eastman, Charles M., "Modeling of Buildings: Evolution and Concepts," *Automation in Construction*, 1, 1992.

[Gero] Gero, J.S. (1990). "Design Prototypes: A Knowledge Representation Schema for Design," *AI Magazine*, 11(4), 26-36.

[ISO] ISO/TC184 (1993). Part 1: Overview and fundamental principles. Industrial automation systems and integration - Product data representation and exchange, Draft International Standard, ISO, Geneva, Switzerland, ISO DIS 10303-1.

[Kunz 94] Kunz, J.C., Fischer, M. A., Levitt, R.E., Teicholz, P.A., "Toward Desktop Engineering," Proceedings of a Workshop in Honor of Steve Fenves, Carnegie-Mellon University, 1994.

[Laseau] Laseau, Paul, *Graphic Thinking for Architects and Designers*," Second Edition, Van Nostrand Reinhold, NY, 1989.

[Luiten] Luiten, G. T., *Computer-Aided Design for Construction in the Building Industry*, Ph. D. thesis, Civil Engineering Department, Delft University of Technology, 1994.

[Luth 91] Luth, G.P., Jain, D., Krawinkler, H., and Law, K.H. (1991). "A Formal Approach to Automating Conceptual Structural Design. Part I-Methodology," *Engineering with Computers*, Vol. 7, No. 2, pp. 79-89.

[Maher] Maher, M. L., and Fenves, S., "HI-RISE: An Expert System for the Preliminary Design of High Rise Buildings," *Knowledge Engineering in Computer-Aided Design*, edited by J. Gero, Elsevier Science Publishers B. V., 1985.

[Milne] Milne, M. "Tools for Climate Responsive Building Design," *Progressive Architecture*, August, 1993.

[Pohl] Pohl, Jens and Chapman, Art, "Expert Systems for Architectural Design, " *Journal of Real Estate Construction*, Singapore University Press, National University of Singapore, 1, 1990, pp. 29-45.

[Schon] Schon, Donald, *The Reflective Practitioner*, Basic Books, NY, 1983.

## *Acknowledgments*

We appreciate the work of Dr. Renate Fruchter in helping to develop early versions of this system. Chris Young of Dillingham Construction Company, Project Manager, provided valuable specification of the scenario. The Center for Integrated Facility Engineering provided financial support for this work, and its members provided continuing encouragement, helpful suggestions, and enthusiastic participation in testing.