# EVALUATION OF AI-BASED CONCEPTS FOR DESIGN

William Bricken
October 1987

## Introduction

The Artificial Intelligence (AI) concepts in this report have been considered for either near or long term integration into the evolving design for a Computer Aided Design System (CADS).

The body of this report contains:

-- a description of seven candidate AI concepts for integration into CADS,

-- an evaluation of the near and long term potential of the candidate concepts,

-- a discussion of the criteria for selection of a single concept for prototype development, and a description of the constraint management system that has been selected, and

-- an overview of the integration of the constraint management tool into CADS.

## Candidate Concepts

A goal of this project is to identify potential extensions of a CADS architecture for inclusion of AI-based computational concepts. It is envisaged that as AI matures as a technology, several of the computational approaches it offers might be integrated into a CADS workstation. The advantages of AI include:

-- advanced interface technologies for human-friendly interaction,

-- concept-based interaction with the computational system, which permits English language-like communication, task specification, and data transaction with the processing system,

-- constraint management and bookkeeping systems that relieve the burden of memory for details and the tedium of repetitive computation from the engineer,

-- automated recording of user interactions, which provides a developmental view of the design process,

-- expert system technology for the coordination of design data and tools, and

-- advanced data structures that permit deductive processing over active objects.

The concepts that we have identified for further exploration and possible prototyping are presented next.

*Concept-based Information Retrieval*

Accessing and extracting information from textual databases, such as the lessons-learned database, is a tedious process. An expert-system permits the structured definition of abstract concepts that can be used to locate ideas in text. Such a tool would provide the designer with a capability to locate information in textual files without resorting to keywords.

*Knowledge-Base Editing System*

Knowledge systems rely on frame-based representations of the knowledge at the user level. A visual display and editing system permits the user to search and locate documents visually, and to edit them as computer files.

*Constraint Management System*

The design process requires the specification of many interacting constraints. A rule-based constraint management system permits the combination of several partially completed designs and imprecisely specified numerical design decisions. Constraints can be intersected automatically to yield the set of possible designs, or to identify contradictions between designs and specifications.

*History and Referencing System*

Design decisions can be automatically recorded and documented as they are made. When designs must be revised, the basis of each decision is available in the record. Common patterns of tool and data usage can be identified, to help with the specification of a general design process.

*Tool Selection and Data Retrieval Expert System*

At the heart of a large automated system is a set of tools and a data interface. An expert system can coordinate requests for tools and data, transforming the output of one tool system into formats that are appropriate for the input of the next system.

*Interface Prototyping System*

Complex designs and displays need to be prototyped. A system that facilitates easy composition and display of interfaces provides a visual prototyping capability.

### *Object Oriented Simulation*

The design process includes simulating design functionality. An object oriented simulation capability permits construction of simulations that are timely and visually informative.

Several other concepts were also reviewed. Most were rejected as candidates because they are highly experimental, and should not be expected to be available technologies within the scope of this project. Among the AI concepts that were found to be too innovative are:

-- models of the designer and the design process,

-- options generation techniques that might enhance the creativity of the designer, and

-- fully automated design.

## Near and Long Term Possibilities

The feasibility of integration of specific AI concepts depends upon the developmental progress of both the design methodology and the AI methodology. Although AI research is far-reaching, *usable* AI implementations of some advanced concepts might not be available. For this project, AI concepts were constrained to technologies that have a reasonable expectation of being delivered as functional systems in the work environment within the next 20 years. Within this ceiling, a *long-term* concept is one that might be expected to be developed successfully with ten years of effort. A *near-term* concept is one that might be expected to be developed with approximately three years of effort.

It might be noted that *expert system technology*, the only AI technology area that has yielded commercial applications of AI techniques, took about fifteen years between development in a university research environment to use in the commercial marketplace. During these years, many of the most talented computer scientists in the US worked on the development effort.

The development effort underlying the implementation of each candidate technology is considered next.

### *Concept-based Information Retrieval*

This AI tool has been developed over the last four years for commercial release, but has not yet been marketed. The textual databases that the tool would address are not yet on-line, although they do exist on paper. The consistency, reliability, common language, abstractness, and other features of the databases have not yet been evaluated.

AI effort: very low

Domain effort: moderate

Development potential: Available with two person-years effort, mostly in domain engineering. Not a critical tool.

### *Knowledge-Base Editing System*

Tools for knowledge base editing have been under wide scale development for two years. The development of this tool assumes that the automation of design process relies substantially on AI.

AI effort: low

Domain effort: high

Development potential: Available with three person-years effort, spread equally between AI, knowledge engineering and system architecture. Critical for AI integration.

### *Constraint Management System*

Constraint management has received moderate attention from the AI community for fifteen years. The technology exists in experimental systems only. Design constraints are specified in standards documents and are used informally by designers. A useful set of constraints would be relatively easy to identify.

AI effort: moderate

Domain effort: moderate

Development potential: Available with three person-years effort, spread equally between AI and knowledge engineering. An extra two person-years required for refinement of domain model. Critical for an automated design system.

*History and Referencing System*

Techniques for recording machine interaction are known, but would need to be tailored for the domain. This tool assumes a large portion of the design process has already been automated.

AI effort: moderate

Domain effort: moderate

Development potential: Available with three person-years effort, mostly in domain automation. A convenient but not critical tool.

*Tool Selection and Data Retrieval Expert System*

The techniques for such a tool are well known, but the specific tool would depend highly on the application tools and databases it was used for. Getting a selection of independent tools to communicate is generally a very difficult task. The design process and a substantial subset of the tools and databases would need to be already automated.

AI effort: high

Domain effort: very high

Development potential: High risk as a general tool; limited availability with indeterminate effort, depending on tool and database selection and system architecture. Allow five person-years. Possibly critical for design automation.

*Interface Prototyping System*

The window and icon technology is understood, although the software tools available are still crude. Models of the design process would have to be in place. This task requires a high degree of integration between the domain and the technology.

AI effort: moderate

Domain effort: high

Development effort: Available with limited applicability with three person-years effort, divided evenly between AI and domain modeling. Requires human factors research. Expect an additional two person-years for enhanced utility.

### *Object Oriented Simulation*

This technology is innovative. Although elements are known, their integration is quite unexplored. A thorough understanding of the automation of the domain is required.

AI effort: very high

Domain effort: very high

Development effort: Depending on the scope of simulation, from five to fifteen person-years. Desirable but unlikely to be available.

## Selection of The Constraint Management System

In addition to development effort, several other criteria for selecting a prototyping concept are relevant. The prototype should be demonstrable rather than just a proof of concept. This implies the prototype should embody a clear functionality, as opposed to being a display with limited capabilities. It is considered appropriate that the tool be capable of being developed without a great reliance on design automation. The tool should also contribute substantially to the needs of the current project, without duplicating efforts already underway.

Using these criteria, the Constraint Management System (CMS) was chosen for prototype development. This system can be prototyped with substantial functionality, without relying on previous automation of design processes. The knowledge engineering necessary to demonstrate constraint management is minimal, since many design specifications are already expressed in terms of constraints. We have decided to focus the prototyping effort on tool functionality rather than on extensive knowledge engineering of the design domain under the assumption that specific aspects of the domain can be configured relatively easily to be used on a fully functional tool.

The CMS offers functionality in a critical aspect of the design process, that of *bookkeeping*. Design requires the balancing and negotiation of hundreds of constraints, many of which interact. The CMS helps to keep track of design decisions and partial decisions. It can alert the designer whenever a particular decision contradicts other decisions or pre-established specifications. It can enforce the effect of decisions on other aspects of the design, and limit the freedom of the other aspects accordingly. Constraints apply early in the design process, and thus it is efficient and cost effective to alert the designer to any violations of constraints. This early warning can save over commitment to designs that might later be found unacceptable. Designs which are mutually incompatible can be identified during parallel development, and thus save redesign effort when the competing designs are being reconciled.

In addition to the CMS prototype, we will furnish a developmental version of a concept-based information retrieval system. We will configure a small set of lessons-learned data in order to demonstrate the capabilities of this system.

## Plans for Integration

Initially, the CMS will be built as a stand alone system. It will be implemented in Common LISP and should be able to run on any workstation that supports Common LISP. We assume that a Common LISP window system will be in place by time of delivery. We expect to develop our version on a SUN 3 workstation.

A constraint database will be developed from design specifications and from interviews with domain experts. This database will also be written in LISP and be stored in the same system as the CMS tool.

In the long term, the CMS might be integrated with the CADS through an attached AI workstation. The CMS would reside on that workstation, while the constraints database would reside within the selected commercial database management system, and be called from the AI workstation. Designs constructed within a CAD tool would need to have their output converted into a format that is compatible with the input expected by the CMS.

If the CMS is treated as a callable tool, each analytic tool would need to have its output expressed or expressible in a form relevant to the constraint model. Analytic models would need to be knowledge engineered into the constraint format, critical points in the design path would need to be identified as needing constraint checking, and tools for the reconciliation of contradictory constraints would need to be developed.

A fully useful constraint management system requires several support tools, many of which have been mentioned as candidate concepts. Specifically, a constraint editing system would be needed for the addition, removal, and change of relevant constraints. And an integrated data retrieval system would be needed to facilitate communication between design data and constraint rules.

## Conclusion

The knowledge engineering task inherent in the automation of design is immense. Not only must the functionality of the design be formally modeled, but also the job of the designer must be modeled. For the current state of the art in expert systems and related AI technologies, knowledge engineering is limited to domains with a well understood technology that can be communicated to a novice within a short time. Thus, it should be expected that automation of the design process is many years away. We might expect that an extended effort will be successful in formalizing major functions of design. However, formalizing the skills of a designer is likely to be impossible within the next twenty years. This observation argues for a mixed-initiative system, for which the difficult tasks can be off-loaded from the computational system to the human designer.

The course of development of the fully automated environment should be segmented. First, localized areas of automation must be identified and implemented. As an example, CAD workstations for the design of three-dimensional geometry represent such an *automation cluster*. Next, each automation cluster must be integrated with AI technologies. In the example, a constraint management system might serve to criticize designs for violation of specifications. Only after several of these *intelligent automation clusters* have been developed and placed in use should an overall integration of design process be attempted. The information gained from experience over several years with automation clusters must feed into the design of the overall integration, because it is only through experience that designers can know the importance and the architecture of each piece. Experience with the automation of a functional cluster should be expected to change the architecture of the integrated system.

To provide an initial prototype of an automation tool, we will develop a constraint management system with broad functionality. This prototype tool should then be used and refined in an application environment, in order to evaluate the effect of such a tool on the design process and the effect of modeling the design domain to fit the formalisms of the tool.

The long term prospects of the integration of AI into the design process depend upon the formalization of design and advancements in AI. Certainly some AI tools that are currently available would have substantial impact on the ease of bookkeeping during the design process. However, their utility depends upon the automation of design data structures, the availability of automated design tools, and the coordination of these tools and databases. Such automated systems do not currently exist for the design process, and their development will be effort intensive. Even after such automated systems are in place, the formalization of the process for application of AI techniques will be extremely expensive in knowledge engineering effort.