Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1998), *12,* 61–63. Printed in the USA.

POSITION PAPER

The computer-aided engineer: Prospects and risks

JAMES H. GARRETT, JR.
Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890, U.S.A.

(Received June 11, 1997; Revised September 22, 1997; Accepted August 8, 1997)

Abstract

Within the last 20 years, much research and development has been conducted to deliver computer tools to assist engineers in performing their tasks. We are now experiencing the power of worldwide networked computing environments and the ability to easily share large quantities of information over geographically dispersed environments. However, are the computing environments available today supporting engineers in doing their job or defining how they must do their job? We need to develop systems that are more transparent and understandable to the users and that are more responsive to the individual needs and idiosyncrasies of the persons using these assistants.

Keywords: Computer-aided Engineering; Intelligent Assistants; Artificial Intelligence

1. INTRODUCTION

Within the last 20 years, much research and development has been conducted to deliver computer tools to assist engineers in performing their tasks. We can model and predict the behaviors of most of the structures that we now build. We now have sophisticated 3D visualizations of that behavior. We have design tools based on Artificial Intelligence (AI) approaches that are able to take a reasonably well-formed problem description and synthesize potential solutions. We are now experiencing the power of worldwide networked computing environments and the ability to easily share large quantities of information over geographically dispersed environments. However, are the computing environments available today supporting engineers in doing their job or defining how they must do their job? What kinds of computing support should we be providing to engineers? I propose that what we should be aiming for is the intelligent assistant. We need to develop systems that are more transparent and understandable to the users and that are more responsive to the individual needs and idiosyncrasies of the persons using these assistants.

Reprint requests to: James H. Garrett, Jr., Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890, U.S.A. Phone: (412) 268-5674; Fax: (412) 268-7813; E-mail: garrett@cmu.edu

2. ARE AUTOMATIC SYSTEMS DESIRED?

There has been a marked shift in the type and level of support expected from computer-aided engineering environments. Early computer-aided engineering (CAE) systems were predominantly analysis tools: for example, given a complete description of the form and the "external loads" put on a structure, the system predicts the structural response. As systems were being developed for design purposes, they were at first, and many still are, focussed on the detailed design and evaluation stages, leaving the early conceptual stages of design unsupported. These detailed design and evaluation tools cannot be used until many of the important cost-defining decisions have already been made. For example, most of the available structural design packages will help optimize a structural system (i.e., find the most efficient member sizes), but do not help in synthesizing the topology of the structural system or in selecting the types of members to use. In addition, such systems usually take control of the solution process after being given a detailed description of the problem, leaving the engineer as an information pre- and postprocessor.

I and a few colleagues once designed and implemented a knowledge-based expert system to design electric power transformers—ENCORE (Garrett & Jain, 1988). This system took a specification for the desired voltage transformations to be provided and the desired efficiencies and temperature rises and then proceeded to explore the space of possible transformer designs. Many pieces of knowledge were applied and many assumptions were made in designing these transformers. Only at the end of the process was the result of this search displayed to the user. The user could not see the various designs considered during the search and could not influence when and which pieces of knowledge were applied. They could not modify their specifications in mid-design or prune branches of the search tree. Such systems, while able to design, did not provide the designers the support they needed.

Knowledge-based systems, such as ENCORE, were advertised as being better than procedural design systems because they made the knowledge transparent to the user. However, this type of knowledge-based design system still did not make the design process transparent and interactive, but rather supported only system-controlled design. The benefits of such knowledge-based systems over traditional procedural systems were more obvious for the developers of such systems than for the users. Smith points out that "professionals cannot, and will not, use automatic systems" because of issues of responsibility and the need to explain results, but also because they like to do design (Smith 1994).

3. INTEGRATING THE DESIGNER INTO SYSTEMS

At Carnegie Mellon, we have been developing a design system that integrates the designer into the solution process, with control of that process lying mostly with the designer. The software environment for the early phases of building design (SEED) supports three design phases: (1) developing a specification of the design problem to be solved (SP); (2) generating a schematic layout of the building in center line form (SL); and (3) generating a schematic three-dimensional model of the building spaces and physical components (SC) (Flemming & Woodbury 1995). In all three of these phases, the design process is modeled as an elaboration of an input specification into a more detailed output specification. For example, the input to SL in the collection of functional units that describe a functional space to be designed and the functional requirements of those spaces, such as minimum and maximum areas and adjacency requirements, and thus compose the architectural program. The output produced by SL is a collection of functional unit-design unit pairs that satisfy the given architectural program. The design units essentially describe the form and location given to the functional units. In all three phases supported by SEED, the designer is given three options for interacting with the design system—fully manual, interactive, and automatic—but what these modes mean differs in each of the three phases. For example, when the SL module is in the fully manual mode, the designer is given complete control as to what functional unit to address and what values get assigned to the geometric attributes of the associated design unit. In the interactive mode of SL, the designer is still able to select the functional unit to design, but the system generates alternative design units, evaluates them, presents the evaluation to the user, and then asks the user to select one. In the automated mode of the SL module, the system selects the sequence in which the functional units are addressed, generates alternative design units for each functional unit, and then evaluates and selects the design units to use based on a built in evaluation function. While the first two modes of the SC module are similar to those of the SL module, the automatic mode of the SC module, because it does not have a built-in evaluation function, still relies on the user to evaluate the generated design units. However, the user can specify that a smaller or larger amount of design to be done between evaluation steps. Most design systems provide only the fully manual mode or the automated mode of operation, but most systems do not offer all three.

4. PROSPECTS AND RISKS

This ability to control a process, but be supported in conducting that process, is what engineers really want; they do not want a black box from which solutions are extracted. Engineers seem to want something like an intelligent assistant from which they can make requests for information; delegate well-defined tasks; ask for ideas while synthesizing various potential solutions to problems; relegate menial, boring, but none the less required tasks, such as code verification, version management, bookkeeping, rationale recording, etc. These intelligent assistants will be expected to learn over time and improve their level of assistance. They should also become oriented to their user and be able to anticipate information needs based on past experience with their users.

Providing a system that is able to follow a designer-controlled design process and then, when requested by that designer, "jump in" and assist in the design process is extremely difficult. It should be noted, however, that this is exactly what earlier automated design systems were requiring of their human users when they encountered a design problem on which they could proceed no further, but then again humans are much more adaptable than computers. The representations used in these more interactive design systems now need to support human and computer processing. The contexts in which processes can and cannot be applied need to be made explicit for each and every process. The knowledge embodied within these processes needs to be made available at a variety of levels of detail known about the design. The alternatives that the computer system can investigate need to be clearly conveyed to the human user, who may wish to prune or augment that list. Such interactive communication is where some difficult system development issues and risks lie.

As we move into the earlier stages of design and attempt to support the user in searching over the space of potential solutions, how do we clearly communicate where that user currently is in that space and where he or she has been? In other words, how do we keep users from getting lost in the design space? This was not a problem when the computer completely controlled the search of the design space, but in the computer-assisted designer-controlled mode, this is a real possibility. How do we keep from overloading the cognitive abilities of the designer, which will surely lead to errors? As a designer is exploring a space, he or she will likely be treated to various streams of asynchronous feedback about various performances of the design, alternative decisions that could be made, recommendations of how the user might better use the system, etc. There are major questions about how, and how well, engineers would make use of such assistance. I agree with Smith that much more experimentation with real human designers is needed to determine what kinds of information and support they really want and can use. Human–computer interfaces will have to be changed to alleviate the complexities and confusions that will most certainly come from using such intelligent design assistants. Computing environments must become more cognizant of the abilities and knowledge states of their users and help them to better use the computing systems. These systems need to recognize that humans are not perfect users of their functionality and should be designed to be error tolerant. The responsibility of proper use should not be placed solely on the shoulders of the user. Excuses such as "the users should have read the manual" or "the users need more training" are simply not realistic in today's multisystem, time-critical work environments.

REFERENCES

Flemming, U., & Woodbury, R. (1995). Software Environment to Support Early Phases in Building Design (SEED): Overview. *J. Architec. Eng. 1*, 147–152.

Garret, J.H., Jr., & Jain, A. (1988). ENCORE: An object-oriented knowledge-based system for transformer design. *AIEDAM 2(2)*, 123–134.

Smith, I.F.C. Designers like designing. *Proc. Int. Workshop on the Future Directions of Computer-Aided Engineering*, 159–163.

James H. Garrett, Jr., Professor of Civil and Environmental Engineering at Carnegie Mellon University, received his B.S.C.E. in 1982, M.S.C.E. in 1983, and Ph.D. in 1986 from Carnegie Mellon University. His research and teaching interests are oriented toward the development of advanced computational methods and technologies for supporting the engineering tasks of analysis, synthesis, and evaluation. He has been investigating representations and processing strategies to support the usage of engineering codes, standards, and specifications. He has also been investigating the application of neural networks to modeling and interpretation problems in structural and environmental engineering.