Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2007), *21*, 11–12. Printed in the USA.
Copyright © 2007 Cambridge University Press 0890-0604/07 $16.00
DOI: 10.1017/S0890060407070059

REFLECTIONS

Engineering design support challenges

IAN F.C. SMITH, Associate Editor
Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

After 20 years of personal experience in introducing advanced software into several engineering contexts (engineering consulting firms, construction companies, large diversified companies, and commercial research establishments), some observations may be of interest. Although some development efforts succeeded and software is still in use, not one development effort has met the expectations that were expressed at the beginning of the project. The least successful experiences are design applications.

As is the case for most events in practical situations, the reasons for poor performance are multiple, diverse, and difficult to enumerate exhaustively. The following list provides a set of partial explanations that have been observed most often:

- Designers liked designing. Even if software allowed designers to work faster, quicker, and better, this was irrelevant because they had to surrender an enjoyable aspect of their work to do so. They invariably did not want to take this step even if it meant that in the end additional support would be beneficial.
- Collaboration between actors in the design required agreement on a common ontology to exchange information. This was never completely successful.
- Validation of the system was not complete because good performance metrics could not be identified.
- The human–computer interface was not sufficiently developed for the needs of engineering designers.
- Engineers in general lacked a fundamental understanding of the concepts of computer science. This lead to inappropriate expectations, bad knowledge modeling, and defective communication with software developers.
- Poor understanding of all aspects of the design task that needed to be supported. For example, when designers had difficulties dealing with key collaborators in other areas of the firm, they did not make this known to the development team. Although the reasons for this may have partly been related to the personalities of those involved, inherent conflicts associated with the task were not uncovered.

Several items above point to research opportunities. For example, much work is needed to develop design-system performance metrics so that proposals can be validated and so that incremental development can result in verifiable improvements. In addition, research into engineer–computer interaction is in its infancy, and much can be done to incorporate more effectively aspects such as domain-dependent symbols, specialized ontologies, multidimensional data plots for exploration, and better manipulation of design constraints and objectives.

There are also more difficult issues that pose more long-term challenges. For example, it does not seem likely that the following issues will be resolved soon:

- Validation of conceptual design support systems. It is very difficult to establish with any reliability that a conceptual design support system helps designers design better than they do without the system. Indeed, many designers claim that computer systems for conceptual work are more likely to hinder their creative design skills.
- Finding the right ontology for all users. Because ontologies contain implicit knowledge that is context dependent, the likelihood that a single ontology is appropriate for all actors in a complex design project is low. For example, this is one of the problems behind the current stalled progress in the use of foundation classes in the construction industry.
- Selecting developers who are also good designers. It can be argued that an important key for success of a design system is if it is developed by someone who has an intimate knowledge of the design task and much experience making integrating designs into practice.

Reprint requests to: Ian F.C. Smith, Ecole Polytechnique Fédérale de Lausanne, Station 18, GC-G1-507, Lausanne 1015, Switzerland. E-mail: ian.smith@epfl.ch

Such combinations of diverse knowledge and capabilities are rare. The majority of computer-aided design researchers have little or no experience in practical design.
- Identifying ways to help designers enjoy designing with computers.

A last issue is education. Over the past 40 years, much computing education in traditional engineering fields, such as mechanical, civil, materials, and chemical engineering, has drifted away from fulfilling any useful purpose. Engineers rarely program in practice, and the programming education they do receive is not enough for most to achieve an acceptable level of competence. The software tools they use in their work are very different from the versions they may have had to struggle with during their undergraduate education. A one-semester programming course followed by software use during project courses does not prepare engineers for their responsibilities in practice. As mentioned earlier, it is also this sort of knowledge deficit that contributes to the lack of practically used design support systems.

Engineers need to have a better understanding of the fundamentals of computer science. For example, all engineers should know that even simple 10-line algorithms may not provide answers for full-size problems even when they are run on the world's fastest computers. Furthermore, there are whole classes of problems that can never be helped by Moore's law. These classes provide fertile ground for development of applications of artificial intelligence in engineering. Other subjects include database design, design space exploration methods, machine learning, and distributed systems.

Engineers need to know why computing is first a science, second an engineering discipline, third and at its lowest level, a skill to be mastered. Most engineers believe that computing is only the last of these three things. These engineers do not know what they do not know. This has important practical consequences. For example, they are unable to see the fundamental principles behind new software products and may not, therefore, benefit fully from their strengths and avoid misuse due to their weaknesses. More importantly, most cannot conceive systems on their own and they are unable to work effectively within teams with computer specialists during software development. From a design support viewpoint this means that the systems currently being proposed have difficulty demonstrating, for example, a positive return on investment. Even less quantitative measures are often not convincing.

To conclude, there are many challenges on the road to providing useful design support for a wide range of engineering tasks. Combined efforts in research and in education are necessary to make progress. The community of readers and contributors to *AI EDAM* are well positioned to take a leading role.