16

DESIGN PROCESS METHODS: DISCUSSION

Research Paradigms

LEO JOSKOWICZ

IBM T. J. Watson Research Center, USA

1. Introduction

Engineering design is a complex, evolving activity, involving many kinds of processes aimed at turning requirements and ideas into products. Concept development, concept evaluation, system-level integration, detailed design, prototyping, manufacturing, production, and marketing, are all part of an increasingly demanding iterative design cycle. Global competition requires shorter and more efficient design cycles, more systematic evaluation of possible solutions, and a better understanding of the relations between the many factors affecting the final product. Faced with these demands, research in computer-aided design has rapidly expanded its scope from traditional drafting programs to systems capable of supporting many of the diverse activities associated with product design.

Understanding and characterizing the different aspects of the design process is the key to the development of a new generation of computer-based design tools. While design methodology has been the subject of study for more than half a century, only recently have formal design process methods emerged from research in CAD. As is the case in other fields, such as finance, communications, and manufacturing, the focus on computational methods has not only provided a better and more systematic foundation to the field, but it has contributed to its evolution, and is likely to give rise to new paradigms.

The goal of this chapter is to briefly reflect on the state of the art of recent research in formal design process methods, identify trends and paradigms, and discuss key technical issues. Most of the background material for this paper is drawn from presentations and discussions at this workshop, although it reflects only the author's opinions and views.

J. S. Gero et al. (eds.), *Advances in Formal Design Methods for CAD*

2. Research Methodology

Most researchers and practitioners view design as a series of interrelated processes that must be studied and characterized to develop adequate CAD tools to support or automate significant aspects of the design activity. These design processes cover a wide spectrum, ranging from the well-defined and highly structured to the poorly understood and difficult to isolate. These processes and the interactions between them are highly dynamic, drawing on different sources of knowledge at different levels of abstraction.

The three major components that have been identified are the design knowledge, the domain knowledge, and the derivation process. The design knowledge is generic, domain-independent, and includes techniques for identifying customer needs, methods for product concept generation, and mathematical optimization, to name a few. The domain knowledge is specific to the type of product being developed and its application domain; it includes part catalogs, physical equations of behavior, and previous designs. The derivation process includes the methods by which new or improved designs are produced, such as transformation and refinement operators, deduction, and induction.

There is a broad consensus about the importance of each of these components. Design methods for the better understood aspects of the design process, such as part catalog selection, collaborative design information exchange, and multi-objective optimization, have been developed and are constantly improving. The current research emphasizes collaborative design methods, conceptual design support, and design rationale capture, among others. Knowledge representation issues related to the context of the design and the explicit representation of form, function, and behavior have also been recently investigated.

A major area of disagreement, addressed explicitly or implicitly in the papers, relates to the nature of the design process itself and the best way to formally represent it. Some researchers propose to use formal logic and postulate induction, deduction, and abduction as the basic design derivation and modification mechanisms. Others take a more empirical stance and attempt to follow established design methodologies as much as possible. Some propose an evolutionary approach and try to reproduce as closely as possible human decision-making. Depending on the approach, deriving properties of the design process can be seen as deriving pragmatic, rule-of-thumb observations, or proving properties of formal systems. These considerations yield very different formalizations.

A related issue is the degree of specificity of the theory with respect to the design domain and process. The question is to determine what is common and what is distinct in architectural design, mechanical design, and VLSI design, to name a few examples. In other words, in what sense is the design of a house similar or different to the design of a car or a household electric appliance? Some fields and design activities are more generative, some emphasize enumeration and selection, while others rely heavily on adaptation of previous designs. Some must follow a highly structured set of rules, while others rely on unquantifiable judgements, such as aesthetics. Yet another difference is whether the design is product-driven (products designed to meet specific consumer needs), technology-driven (products designed to push a new technology), or process-driven (products whose characteristics rely heavily in a production process). These differences are not merely a question of the degree of creative, innovative, detailed, or routine design involved: they are at the base of the discipline and the design process itself.

These issues have a major impact on how researchers develop formal and computational theories of the design process. The definition, systematization, and exploration of the design space, its focus and representation, and the derivation processes heavily depend on the genericity and domain-independent assumptions that are made. This helps to explain why some works emphasize mathematical formalization (logic, optimization, etc.), while others take a procedural or behavioral approach, focusing on reproducing the input/output behavior of the human design process. Approaches that attempt to mimic human designers are based on empirical studies of the design process through user interviews and design protocols.

This diversity in assumptions, emphasis, and focus makes comparing and evaluating the proposed methods very difficult. Some researchers have proposed to use benchmarks as an objective means to compare different approaches. Benchmark studies are indeed useful to understand the scope, coverage, and efficiency of a set of methods with respect to a task. However, they are only suited to relatively mature fields, with well-defined and well-understood problems, such as comparing the performance of two computers. For design process methods, we cannot even begin to agree on what the problem is, let alone define what criteria should be used to compare them.

3. Design Process Paradigms

In reviewing current work on design process methods, it is interesting to note that most of the proposed methods are build around a single paradigm, with the assumptions, simplifications, and specific focus (not always explicitly stated) that this entails. I will briefly review the two papers presented on design process methods. The goal is to identify the paradigm upon which they are based and make their assumptions explicit.

Brazier, Van Langen, and Treur present a logical theory of design based on formal semantics for both the static (knowledge) and dynamic (reasoning) aspects of design. The theory, which is based on many-sorted first order predicate logic, partial temporal models, and non-monotonic reasoning, is generic and domain independent. The authors justify their approach with the paradigmatic observation:

> "Design tasks typically reason with incomplete and inconsistent knowledge of requirements and design object descriptions; they reason non-monotonically

> with and about, for example, (default) assumptions, contradictory information, and new design knowledge."

Interestingly, the authors directly go on to describe the technical details of the formal semantics, which they claim provides a means to model design strategies, without explaining why such a formal semantics is desirable.

Lei, Taura, and Numata present a product model-oriented approach, where the data associated with the product at the different stages of the product life cycle serves as the key organizing principle. The model supports the generation, exploration and navigation of the design objective in each task or activity, and the evolution and alternatives in the design history. This paradigm is summarized by the following observation:

> "The collaborative design process can be viewed essentially as the evolution of product data, which is the result of a series of decisions."

The central role of explicit modeling of the evolution alternatives and constraints of the product data justifies the emphasis on the product data model. The computer implementation, based on the language STEP (Standard for the Exchange of Product Model Data) targets electro-mechanical product design.

4. Conclusion

As the reader can appreciate, it is difficult to compare or even put these theories in perspective with respect to each other. In my opinion, there should be a greater effort to precisely characterize the scope and domain of application of each theory, and identify both their theoretical and pragmatic limitations, which are not always the same. For example, a theory can be very expressive but practically very difficult to compute with, or vice-versa. The critical assumptions must be stated explicitly, and greater effort should be put in clearly stating the paradigm upon which it is based, the domain where it is most likely to be useful, and the domain or type of design where it is not useful.

Following this methodology will allow us to begin assembling a "toolbox" of design methods, while the "ultimate" theory (if such a theory indeed exists) is being worked on. It will allow us to identify areas that require more research, and problems that deserve more attention. It will provide practitioners with an understanding of what is and is not presently possible, and useful approaches, if not programs, to address their problems.

Conversely, and on a more pragmatic level, we should also examine existing needs and identify the current bottlenecks in the design of a particular class of products and domains. This will help us identify what developments would yield the maximum benefit, and help motivate research both in the short and the long term.