ADVANCES IN FORMAL DESIGN METHODS FOR COMPUTER-AIDED DESIGN

JOHN GERO
University of Sydney, Australia

1. Introduction

The introduction to the closing chapter of the proceedings of the workshop on Formal Design Methods for Computer-Aided Design (Gero, 1994) commences with the following paragraphs, which are still applicable.

The primary axiom of formal design methods is that design is a process, ie, it is temporally based and various distinguishable activities can be ascribed as occurring during the process. The use of formal design methods does not imply that design, when carried out by humans, is based on these formal methods. Rather, the use of formal methods initially provides a framework for our notions about design. Just as the use of scientific methods to study emotion does not make emotion a scientific process, so the use of scientific methods to study design does not make design a scientific process. However, once we have formal design methods we can conceive of uses for them other than as simple descriptive devices.

What are the possible roles that formal methods can play in design and in computer-aided design in particular? Formal methods in other disciplines have been found to be useful in a variety of ways and, by analogy, we are able to enunciate roles for formal methods in design. An incomplete list of potential roles includes:

1. informing us about design as a process;
2. providing a framework for comparisons amongst alternate design processes; and
3. providing a basis for the development of design tools.

To this list we can add the following:

4. provide a basis for distinguishing different kinds of design processes.

2. Informing Us About Design As a Process

Design is considered to amongst the most complex and most intellectual of human activities. It is the basis for the change of the physical world we inhabit. As such it is surprising that it is neither well understood nor well characterised.

Design is widely regarded as a process during which the designer carries out distinguishable activities. These activities can be characterised through a formal structure which can be used to provide a framework to assist in the understanding of what design is. Introspection, retrospection and protocol (Gero and McNeill, 1996) studies provide a basis for this use of formal methods. A formal method is proposed as a structure capable of providing a framework for our ideas about design. That structure is often found to be satisfactory in providing a framework for some aspects only of our understanding of design. However, other aspects may not be accounted for, hence other structures are needed to augment or replace the existing ones.

Formal design methods provide the opportunity to characterise design as a process in a uniform manner using concepts and terminology which transcend the individual designer and gives primacy to the processes of designing. Formal design methods are often taught as part of either a design theory course or in a design course as a process of designing. There appears to be a divide between the technology-oriented design courses such as in engineering and those in the human-oriented design courses such as in industrial design and to a lesser extent architecture. The former courses are beginning to embrace formal design methods whilst the latter less so.

Early formal methods proposed the analysis-synthesis-evaluation model of design. Whilst today this is considered to be an inadequate model it has provided the framework for the development of many other process models. Other characterisations, such as that of function-behaviour-structure, of routine/non-routine design, and of case-based/compiled knowledge design have all added to our armory to aid our understanding of design. What these formal characterisations have shown is that design is a highly complex activity which needs many characterisations for its understanding.

3. Framework For Comparisons Amongst Alternate Design Processes

The field of design research is populated by researchers who come from very varied backgrounds. They come from the design disciplines themselves: architects, engineers, industrial designers, and so on. They also come from such disciplines as psychology, computer science, history and philosophy. Given this variegated background one role of formal methods is to provide a framework which allows comparisons to be made amongst alternate design

processes or alternate descriptions of the same design processes. It is not intended here to provide a framework which allows for the comparison of alternate design processes. A few examples will suffice to illustrate the ideas involved. Whilst a model of design based on concepts from optimization appears to bear little relation to one based on constraint satisfaction, they can both be categorised under a formal approach based on search. The notion of search, itself, carries with it concepts associated with a space to be searched. This introduces the need to characterise design spaces and provides the opportunity to extend our understanding of what design spaces are and how they may be created and examined.

Whilst there are well-defined frameworks derived from both within and without design research, none appears to be able to provide both the breadth of abstractions and the depth of process structure needed to be all encompassing. The *function-behavior-structure* framework (Gero, 1990; Umeda et al., 1990) has proven to be both useful and durable as a means of articulating fundamental differences in the characterizations of design activity unrelated to individual processes which execute that activity.

The other common framework which is process-oriented is the *search-exploration* framework. Here search is considered to be that set of design processes which assumes that design can be treated as operating within a fixed space of possible designs and any design process can be characterized as searching for appropriate solutions. From a computational viewpoint this is a very attractive approach since it readily maps onto various standard algorithms such as numerical optimization in the numerical world and constraint-based reasoning in the symbolic world. On the other hand, exploration is that set of design processes which manipulate design spaces which are then subsequently searched.

Both these frameworks are suitable at an abstract level for providing a structure for alternate design processes but are not concrete enough to allow any form of detailed comparison. The function-behavior-structure framework is useful in determining the locus of a process within a design activity since it articulates the six, common different design activities:

1. design formulation
2. design synthesis
3. design analysis
4. design evaluation
5. design reformulation
6. design documentation.

The primary advance in this area has been the increasing recognition that the search-exploration framework provides a means of distinguishing design from many other activities which it appears to look like; activities such as problem solving and planning. The effect of this has been to allow such

design processes as case-based design and a variety of 'creative' design processes to fit into this framework in order to make comparisons between them at this level of abstraction. This comparison is not via benchmarking but in terms of locus in the design process and in terms of the capability of the process to produce the stated output.

4. Basis for the Development of Design Tools

There are two complementary views taken about formal design processes. The first is that formalizing such processes helps us understand design as a series of processes better. The second is that formal design processes are the basis for the development of design tools. A design tool is a computer program which takes a specified design task and provides one of three levels of design aid:

1. provides active support in the form of a complete solution to that design task, for example synthesising or selecting a particular component which satisfies a set of constraints;
2. provides active support in the form of the beginnings of a solution to that design task, for example a conjecture about a possible direction or path to follow; or
3. provides passive support for a design task, for example an analysis or evaluation related to that design task.

This workshop has demonstrated that a variety of novel processes can be introduced into design.

5. Distinguishing Different Kinds of Design Processes

For many computer-aided design simply refers to the process of documenting a design, thus for most practitioners today CAD stands for either computer-aided drafting or computer-aided analysis although there is an increasing realization that there is more. Formal approaches allow us to distinguish different kinds of design processes according to both their locus in an overall view of design and their methodology.

There appear to be three bases of design methods:

1. those based on empirical results;
2. those based on axioms; and
3. those based on conjectures.

This third category can be further subdivided into:

- 3.1 those where the conjecture is founded on an analogy with human design processes, and

3.2 those where the conjecture is founded on concepts other than human design processes.

Surprisingly, there are very few empirical results derived from human designers on which to base a design method. As a consequence there are very few formal design methods which are directly based on human design processes. Just as there very few design theories to explain the phenomena of human designing. The closest example of this class of design methods is that presented by Rodgers et al. in this volume although it is not strictly based on direct empirical evidence.

Formal design methods based on axioms are more common than those based on empirical results but even here there are remarkably few and the discussion by Rudolph in this volume examines some of the reasons why there are so few. Such axiom-based formal methods include those presented by Rudolph and by Brazier et al. in this volume.

By are the most common basis for current formal design methods are conjectures, conjectures of both kinds listed above. Design by combination is a good example of a formal design method based on a conjecture which is itself founded on an analogy with a human design process which is claimed to be of that kind. This class of design methods is the most common even though not all the conjectures are well-founded. Amongst this class is found the formal methods of: design by generate-and-test, design by refinement, design by analogy, case-based design, top-down methods, etc. Examples of this class of formal design methods in this volume include those presented by Kalay and Carrara, and Grabowski et al.

The second class of conjecture-based design methods include the mathematically-based methods and the large corpus of design systems which use a heuristically derived system architecture as the framework. Examples of this class of formal design methods in this volume include those presented by Maher et al., Gero and Kazakov, Brown and Cagan, Zozaya-Goristiza and Estrada, Lenart, and Lei et al.

6. Conclusions

Formalizing design methods allows them to be better understood before they are used. It provides the basis for any mode of comparison between methods whether these comparisons be based on the methods' complexity, ontology, epistemology, teleology or applicability. It becomes possible to examine the method separately from the results it produces and to determine its behavior. It becomes possible to consider a method's expressive power and to determine whether it is a strong but specialised method or a weak but general method.

Perhaps the most significant aspect of formalizing design methods is that it then becomes possible to implement them as computer programs which radically changes both their character, their testability and their applicability. Without such formalization computerization becomes extremely difficult if not impossible. Computer implementations also allow for methods to be tested empirically on a wide variety of design problems.

The eleven contributions found in this volume provide a guide to the state-of-the-art of formal design methods for computer-aided design. They demonstrate both the range and depth of such methods. They also demonstrate that formalizing design methods is still in its early stages and more needs to be done to derive all the benefits from such formalizing.

References

- Gero, J. S.: 1990, Design prototypes: A knowledge representation schema for design, *AI Magazine* 11 ((4): 26-36.
- Gero, J. S.; 1994, Formal design methods for computer-aided design, in J. S. Gero and E. Tyugu (eds), *Formal Design Methods for Computer-Aided Design*, North-Holland, Amsterdam, pp. 353-359.
- Gero, J. S. and McNeill, T.: 1996, An approach to the analysis of design protocols, *Research in Engineering Design* (submitted).
- Umeda, Y., Takeda, H., Tomiyama, T. and Yoshikawa, H.: 1990, Function, behavior and structure, in J. S. Gero (ed.), *Applications of Artificial Intelligence in Engineering V: Design*, Springer-Verlag, Berlin, pp. 177-193.