Design Research: Empirical, Foundational and Developmental Perspectives

Frances Brazier and Jan Treur
Vrije Universiteit Amsterdam
Department of Mathematics and Computer Science
Artificial Intelligence Group

1. INTRODUCTION

Recent research within the Artificial Intelligence Group of the Department of Mathematics and Computer Science at the Vrije Universiteit Amsterdam in the field of AI and Design has focussed on the development of a compositional framework for the design and development of design support systems. Practical experience with the design and development cycle (task analysis, modelling, design, formal specification and implementation) within diverse domains has increased insight in the complexity of design and of the theoretical foundations required.

Complex reasoning patterns have been analysed and formally specified within the design framework, determining the formal semantics of the resulting behaviour of the system and interaction between systems and users. The integration of various techniques, both knowledge-based and non-knowledge-based, is clearly essential for design. The compositional approach to system design facilitates such integration, both within systems, but also between systems and systems and users. Techniques for the specification of interaction between systems, and between systems and users have been explored and incorporated within the framework, although further research is clearly warranted. This also holds for the reuse of models and specifications: a number of possibilities have been explored and tested. Reuse of the framework for redesign is one of the current focuses of research within which compositional architectures are the object of redesign.

In general our research can be viewed from three related perspectives: an empirical perspective, a foundational perspective and a developmental perspective. The integration of these perspectives is the challenge which we face.

A *Empirical perspective*: Close analysis of design processes as required and performed by designers in cooperation with design support systems.

B *Foundational perspective*: Development of logical theories covering both the static aspects and dynamic aspects of design processes.

C *Developmental perspective*: The perspective of the developer of design systems and the support provided to him or her: modelling/specification languages, (task) models for design, implementation environments, et cetera.

The three perspectives are clearly intertwined. Design theories are based on design practice: either on current practice, or on current recognition of limitations for which solutions need to be found. Logical design theories (in particular) are used to define the formal semantics of both static and dynamic aspects of design: providing a basis for verification and validation, and for techniques supporting system design.

2. EMPIRICAL PERSPECTIVE

Practical experience in design support system design is an essential element within our research, providing insight in the design process. A wide variety of design tasks have been modelled and specified:

- elevator design	(Sisyphus'94)
- chemical process design	(AKZO)
- financial portfolio design	(SKBS, Rabobank, ING)
- environmental policy design	(SKBS, RIVM)
- office plan design	(Sisyphus'91/'92)
- financial routing design	(SKBS, Rabobank)
- emission inventory design	(SKBS, TNO)

The following issues have been addressed:

- identification of (conceptual) *design methods, models* and *strategies* (to be) used for analysis of design processes
- analysis of levels of *cooperation* and *interaction* between designers and design support systems
- development of techniques for *verification* and *validation* of design support systems

Each of these issues will be discussed below in more detail.

2.1 Design methods, models and strategies

To analyse design processes conceptual models for design tasks at different levels of genericity have been developed. These models include at least knowledge of:

1 the tasks and subtasks involved in design (*task decomposition*)
2 the way in which the tasks are controlled (*control decomposition*)
3 the domain knowledge (*knowledge structures*), including multiple classifications (*views*)
4 the agents involved in a design process (*role delegation*)
5 the *information links* between tasks and agents

The design method and models include formal specification of static and dynamic aspects of design processes such as:

- non-monotonic reasoning, the use of (default) assumptions, preferences, re-vision,
- constraints and requirement qualifications,
- the acquisition of design requirements,
- incomplete, uncertain and inconsistent information,
- reflective reasoning processes,
- the role and form of history and design rationale,
- identification and retrieval of known cases,
- strategic knowledge and control,
- interaction between agents (designers, design support systems) in cooperation,
- identifying and describing various forms of strategic knowledge and reasoning,
- dynamics of problem statement and (soft and hard) requirements,
- representation, integration and coordination of views,
- representation, integration and coordination of agents.

2.2. **Models of cooperation and interaction**

Within the context of a given design task often specific subtasks may be assigned to either a designer or the system. For example, a design support system may ask the designer for a value for an attribute of the design object that is yet undefined in the system. This type of interaction, object level interaction, in which one of the parties (often the designer) is requested to provide facts of this type, is not uncommon to design support systems.

Interaction between designers and design support systems is, however, often of a slightly different nature. In design processes designers frequently wish to influence the factors on which designs/decisions are based: the goals, the heuristics employed, preferences, assumptions, using the system to explore the results of different strategies. Interaction at this level, the level of strategic preferences, is not uncommon within design, but is not often included in design support system design.

Although a shared task model on which a design support system has been designed is the result of interaction with designers, it is not necessarily the correct model of the design task. A designer may want to be able to influence, for example, the sequencing or choice of subtasks in a particular situation. The design support system with which a designer interacts should make this possible. This is not only of importance for the individual designer, but also for other designers (often the designer(s) involved in the design of a system represent a class of designers) for which the model can be seen as a model of consensus. This model may need to be adapted for individual designers. This level of interaction has been termed the level of task model modification.

To model the knowledge required at these three levels of interaction within a task model, a task based framework for design is required and appropriate forms of representations. It is often unclear not only which representations are best suited to the particular task at hand, but also which differences exist between preferred representations between designers. Some work on this area is reported in (Brazier & Treur, 1994; Brazier, Treur & Wijngaards, 1994).

3. FOUNDATIONAL PERSPECTIVE

Within our research design is viewed as a complex reasoning process within which different types of reasoning are entailed at different levels of concreteness such as: deductive (e.g., to infer properties of a design object on the basis of known properties), abductive reasoning (e.g., to assert properties to meet given requirements of the design object).

A substantial amount of research has focussed on defining models of design as a basis for the development of design support systems, without considering the underlying logical structure of these models. Although some research on AI and design includes logical descriptions of design systems, a logical framework such as the framework on which our research is focussed, that covers both the static and dynamic aspects of design processes has yet to be devised. A logical framework enables formal specification of domain-specific design systems and development of supporting tools for verification and validation.

3.1 Requirements for a logical design framework

A logical framework should provide (see also the list in A.):

- logical representation of (incomplete) descriptions of objects
- a logical description of views
- means to formulate alternative requirement sets and to qualify these possible alternatives (e.g., preferences, possible relaxations of an initial requirement set, new requirements)
- means to formulate a problem statement
- flexibility to describe the dynamics of alternating steps made within and between the problem statement formulation space, the requirement qualification set space and the design object description space

- possibilities for defining design strategies based on different methods and coordination applied to the different spaces (including problem statement formulation, requirement qualification, design object description)
- a description of the complex patterns of (nonmonotonic) reasoning involved in design
- (multi-agent) cooperation

3.2 **Semantics of reasoning behaviour**
In recent research, a logical foundation of the dynamics of complex reasoning patterns has been based on:
1 the notion of representing *information states* as partial models defined in many-sorted predicate logic (Langen & Treur, 1989);
2 the notion of a *functionality description* as a mapping from partial models to partial models (Treur, 1991b);
3 the notion of *reasoning trace* as a sequence of partial models (or partial temporal model) acquired by well-defined transition functions between partial models (Treur, 1992).
This approach has been successfully applied in a number of areas: to define a formal semantics for (the dynamics of) meta-level architectures based on temporal logic (Treur, 1992); to obtain semantics for default logic including the dynamics of default reasoning (Engelfriet & Treur; 1993, 1994); and to obtain a formal model for the dynamics of compositional reasoning systems (Gavrila & Treur, 1994). Application of this approach to design is one of the current focuses of research; in (Brazier, Langen & Treur, 1994) a first attempt is made to define a logical foundation incorporating dynamic aspects.

3.3 **Techniques for verification and validation**
During system design, models and representations need to be continuously verified and validated. The empirical basis for decisions taken during the design of a design support system needs to be considered. A formal framework has an important added value, namely that it opens a perspective to establish (verify) properties of design systems and to develop automated tools for verification and validation. Generic work on validation and verification of static and dynamic aspects of compositional reasoning systems is reported in (Treur & Willems, 1994ab). This work has yet to be specialised to design support systems.

4. DEVELOPMENTAL PERSPECTIVE

Modelling (re)design entails modelling the domain (i.e., the world of interest), the requirements for each of the parties involved, the design objects, and the design process. Our research aims at modelling design tasks within a formal framework for the specification of complex tasks.

4.1 Declarative specification of dynamics

This approach is based on a (multi-level) logical analysis of complex reasoning tasks, using the notion of a meta-level architecture (Maes & Nardi, 1988), and has been discussed in publications on both fundamental and applied research (e.g., Langevelde et al., 1992). In a meta-level architecture, it is possible to reason not only with relations in order to infer properties of, for example, design objects (object-level reasoning), but also about such relations (meta-level reasoning) and (the control of) the dynamics of the reasoning pattern.

4.2 Compositional architectures and formal specification

Modularity in system design is a well recognised requirement for efficient and effective software development, maintenance and re-usability. This is especially true for design support systems. Within current research, the implications of modular design for the design of such design support systems (based on meta-level architectures), are translated into specifications for a flexible environment for the development of modular design support systems. The transparent architecture of the environment in which system developers are provided re-usable modules for both knowledge-based reasoning and conventional components, is a fundamental concern. Design tasks require both knowledge-based reasoning modules and conventional modules such as databases, calculation (or simulation) modules and optimisation algorithms (OR). The integration of such modules in the design and implementation of integrated complex systems for specific design tasks provide both (1) task-specific building blocks and tools and (2) generic task models. The task-specific tools (for example, shells) and generic task specifications combined, provide the basis for well-structured, transparent, modular system development. The emphasis in our current research is on the phase of design, specification and implementation of design support systems. Formal specifications can be transformed into prototype implementations automatically. The employment of automated tools not only significantly decreases the time needed to develop a prototype, but also increases maintainability.

4.3 Generic task models

Within our framework task-specific building blocks, tools and techniques for design tasks have been devised on the basis of experience. Within this framework for realistic design tasks the integration of both quantitative and qualitative techniques is essential; for example, in some situations search and optimisation techniques and constraint manipulation need to be integrated with knowledge-based reasoning modules. The integration requires fundamental research and is thus being studied within this context.

5. REFERENCES

Alberts, L.K., P.M. Wognum & N.J.I. Mars (1992). Structuring Design Knowledge on the Basis of Generic Components. In: Gero, J.S. (ed.), *Artificial Intelligence in Design '92, Proceedings AID '92*. Kluwer Academic Publishers, Dordrecht, pp. 639-656.

Brazier, F.M.T., P.H.G. van Langen, Zs. Ruttkay & J. Treur (1993). On the Formalization of Design Processes. In: Faltings, B. (ed.), *Proceedings of the IJCAI '93 Workshop on AI in Design*, Chambéry, pp. 78-88.

Brazier, F.M.T., P.H.G. van Langen, Zs. Ruttkay & J. Treur (1994). On Formal Specification of Design Tasks. In: Gero, J.S. (ed.), *Artificial Intelligence in Design '94, Proceedings AID '94*. Kluwer Academic Publishers, Dordrecht, pp. 535-552.

Brazier, F.M.T. & J. Treur (1994), User-centered knowledge-based system design: a formal modelling approach, in: L. Steels, G. Scheiber, W. van de Velde (eds.), A future for knowledge acquisition, Lecture Notes in AI, vol. 867, pp. 283-302

Brazier, F.M.T., J. Treur, N. Wijngaards (1994), Modelling interaction with experts: the role of a shared task model, Report, Vrije Universiteit Amsterdam, Department of Mathematics and Computer Science

Brown, D.C. & B. Chandrasekaran (1989). *Design Problem Solving; Knowledge Structures and Control Strategies*. Pitman, London.

Brumsen, H.A., J.H.M. Pannekeet & J. Treur (1992). A Compositional Knowledge-Based Architecture Modelling Process Aspects of Design Tasks. In: *Proceedings of the 12th International Conference on Artificial Intelligence, Expert systems and Natural Language, Avignon-92*. Vol. 1, pp. 283-294.

Chandrasekaran, B. (1990). *Design Problem Solving: a Task Analysis*. AI Magazine 11 (4), Winter 1990, pp. 59-71.

Coyne, R.D., M.A. Rosenman, A.D. Radford, M. Balachandran & J.S. Gero (1990). *Knowledge-Based Design Systems*. Addison-Wesley Publishing Company. ISBN 0-201-10381-8.

Engelfriet, J. & J. Treur (1993). A Temporal Model Theory for Default Logic. In: Clarke, M., R. Kruse & S. Moral (eds.), *Proceedings of the 2nd European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, ECSQARU '93*. Springer Verlag, pp. 91-96. Extended version: *Technical Report IR-334*. Vrije Universiteit Amsterdam, Department of Mathematics and Computer Science, 1993, pp. 38.

Engelfriet, J. & J. Treur (1994). Temporal Theories of Reasoning. In: C. MacNish, D. Pearce, L.M. Pereira (eds.), Logics in Artificial Intelligence, *Proceedings of the 4th European Workshop on Logics in Artificial Intelligence, JELIA '94*. Lecture Notes in AI, vol. 838, Springer Verlag, pp. 279-299. Extended version in: Journal of Applied Non-Classical Logics, Special Issue with selected papers from JELIA'94, to appear, 1995

Finger, M. & D. Gabbay (1992). *Adding a Temporal Dimension to a Logic System*. Journal of Logic, Language and Information 1, pp. 203-233.

French, R. & J. Mostow (1985). *Toward Better Models of the Design Process*. AI Magazine 6 (1), pp. 44-57.

Gavrila, I.S. & J. Treur (1993). A Formal Model for the Dynamics of Compositional Reasoning Systems. *Technical Report IR-323*, Vrije Universiteit Amsterdam, Department of Mathematics and Computer Science. Shorter version in: A.G. Cohn (ed.), *Proceedings of the 11th European Conference on Artificial Intelligence, ECAI '94*. John Wiley & Sons, Chichester, 1994.

Gero, J.S. (1990). *Design Prototypes: a Knowledge Representation Schema for Design*. AI Magazine 11 (4), Winter 1990, pp. 26-36.

Langen, P.H.G. van & J. Treur (1989). Representing World Situations and Information States by Many-Sorted Partial Models. *Technical Report PE8904*, University of Amsterdam, Department of Mathematics and Computer Science.

Langholm, T. (1988). *Partiality, Truth and Persistence*. CSLI Lecture Notes, No. 15. Stanford University, Stanford.

Logan, B.S., D.W. Corne & T. Smithers (1992). Enduring Support: On Defeasible Reasoning in Design Support Systems. In: Gero, J.S. (ed.), *Artificial Intelligence in Design '92, Proceedings AID '92*. Kluwer Academic Publishers, Dordrecht, pp. 433-454.

Takeda, H., P.J. Veerkamp, T. Tomiyama & H. Yoshikawa (1990). *Modelling Design Processes*. AI Magazine 11 (4), pp. 37-48.

Tham, K.W. & J.S. Gero (1992). PROBER—A Design System Based on Design Prototypes. In: Gero, J.S. (ed.), *Artificial Intelligence in Design '92, Proceedings AID '92*. Kluwer Academic Publishers, Dordrecht, pp. 657-675.

Tomiyama, T. & H. Yoshikawa (1987). Extended General Design Theory. In: Yoshikawa, H. & E.A. Warman (eds), *Proceedings of the IFIP WG 5.2 Working Conference on Design Theory for CAD*. North-Holland, pp. 95-125.

Treur, J. (1989). *A Logical Analysis of Design Tasks for Expert Systems*. International Journal of Expert Systems 2, pp. 233-253.

Treur, J. (1991a). A Logical Framework for Design Processes. In: Hagen, P.J.W. ten, and P.J. Veerkamp (eds.), *Intelligent CAD-systems III, Proceedings of the 3rd Eurographics Workshop on Intelligent CAD systems*. Springer Verlag, pp. 3-20.

Treur, J. (1991b). Declarative Functionality Descriptions of Interactive Reasoning Modules. In: Boley, H. and M.M. Richter (eds.), *Processing Declarative Knowledge, Proceedings of the International Workshop PDK-91*. Lecture Notes in AI, vol. 567. Springer Verlag, pp. 221-236.

Treur, J. (1992). Towards Dynamic and Temporal Semantics of Meta-Level Architectures. *Technical Report IR-321*. Vrije Universiteit Amsterdam, Department of Mathematics and Computer Science, Amsterdam. Shorter version: Temporal Semantics of Meta-Level Architectures for Dynamic Control of Reasoning, in: F. Turini (ed.), *Proceedings of the 4th International Workshop on Meta-Programming in Logic, META '94*. Lecture Notes in Computer Science, vol. 883, Springer Verlag, 1994.

Treur, J. & P.J. Veerkamp (1992). Explicit Representation of Design Process Knowledge. In: Gero, J.S. (ed.), *Artificial Intelligence in Design '92, Proceedings AID '92*. Kluwer Academic Publishers, Dordrecht, pp. 677-696.

Treur, J. & Willems, M. (1994a), A logical foundation for verification, in: A.G. Cohn (ed.), Proc. ECAI'94, Wiley & Sons, pp. 745-749

Treur, J. & Willems, M. (1994b), On verification in compositional knowledge-based systems, in: A. Preece (ed.), Proc. ECAI'94 Workshop on Validation of knowledge-based systems, pp. 4-20

Vescovi, M. & Y. Iwasaki (1993). Device Design as Functional and Structural Refinement. In: Faltings, B. (ed.), *Proceedings of the IJCAI '93 Workshop on AI in Design*, Chambéry, pp. 55-60.