The Knowledge Engineering Review, Vol. 20:3, 271–276. © 2006, Cambridge University Press
doi:10.1017/S0269888906000609 Printed in the United Kingdom

Design, innovation and case-based reasoning

ASHOK K. GOEL[1] and SUSAN CRAW[2]

[1]*College of Computing, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332-0280, USA;*
e-mail: ashok.goel@cc.gatech.edu
[2]*School of Computing, The Robert Gordon University, Aberdeen AB25 1HG, Scotland, UK;*
e-mail: S.Craw@comp.rgu.ac.uk

Abstract

The design task is especially appropriate for applying, integrating, exploring and pushing the boundaries of case-based reasoning. In this paper, we briefly review the challenges that design poses for case-based reasoning and survey research on case-based design ranging from early explorations to more recent work on innovative design. We also summarize the theoretical contributions this research has made to case-based reasoning itself.

1 Introduction

'All design is redesign' has long been a popular cliche in design research. In fact, many design researchers developed theories of design based on this theme even before the invention of case-based reasoning (CBR) in artificial intelligence. To take just one example, in his seminal book, Alexander (1964) noted that designs of new villages in rural India are almost always tweaks, modifications or adaptations of existing village designs. He also analyzed the similarities among the designs in terms of design patterns. Thus, the design task is especially appropriate for applying, integrating, exploring and pushing the boundaries of CBR.

Eight factors, however, conspire to make the design task very challenging for CBR.

Breadth Design pertains to all kinds of artifacts. The domain of the design task is extremely wide-ranging. Practical design domains in which CBR has been explored include architecture, engineering, computer software and human–machine interfaces. Also, within a domain such as engineering, design sub-domains range from civil engineering to chemical engineering to aerospace engineering to industrial engineering. Furthermore, even within a design sub-domain such as industrial engineering, design outputs range from designs of products and systems, to designs of processes and services.

Stages Design involves a multitude of stages. The design process starts with some design requirements, and ends in a realization of a product that satisfies the requirements. Some of the common stages of product design are requirements analysis, preliminary design, detailed design, geometric modeling, simulation, optimization, embodiment, prototype testing, manufacturing and assembly. CBR so far has been explored largely for preliminary (or conceptual) design.

Specification Design is very open-ended. In general, the specification of the design problem may evolve during the design process. That is, the problem and the solution specification may co-evolve. For example, a design problem may be underspecified and its specification may need to be completed as part of the design process. As another example, a design problem may be over-constrained and some constraints may need to be relaxed to find a satisfactory design solution.

Complexity Design is extremely complex. It generally involves a large number of interacting components. Even in a simple problem in product design there can be dozens or even hundreds of components. In the design of a complex system there can be thousands, tens of thousands or even millions of components. CBR so far has been explored for designing products in which the number of components is in the dozens.

Collaboration In practice, large-scale design typically is performed by teams of designers. A design team may be distributed both spatially and temporally. Thus, not only is the design team situated in a given social and cultural context, but also each individual designer works in the context of the team.

Representation Design typically involves consideration of both form and function. External representations of designs are usually multi-modal. The form of a design typically is represented as a two-dimensional (2-d) drawing (sketch or diagram) or as a three-dimensional (3-d) solid model. In contrast, functions and behaviors are typically represented using symbolic structures such as schemas and graphs, and performance is typically represented using numerical measures.

Integration There already exist a number of computer-aided design (CAD) tools and environments. Thus, in order to have impact on design practice, CBR tools for supporting interactive design need to be integrated with the CAD tools and embodied in the CAD environments.

Creativity While most everyday design is routine and mundane, much of interesting design is innovative and creative. Thus, on one hand, the design task in general poses the challenge of addressing the issues of innovation and creativity, and, on the other, it offers an opportunity to explore these issues.

It follows from the above that, at present, there is no real case-based theory of design, let alone any case-based technology or tool for supporting design in practice. Instead, research on case-based design so far represents only preliminary explorations into a vast space. In this summary, we briefly review some of the major themes in case-based design research, but note that this review neither is, nor intended to be, exhaustive. Instead of trying to cover all related work, we focus on long lines of research that have not only explored CBR in design but also contributed to CBR in return. Maher & Pu (1997) provide details of some of the ideas and techniques described below.

2 Early explorations

Navinchandra's (1991) CYCLOPS was the first system to explore CBR in interactive design. It provided landscape architects with access to annotated landscape drawings, where the annotations on a drawing explained some of the design decisions in the past case. However, CYCLOPS left the tasks of case adaptation and design evaluation to the architect. It was followed by CADET (Sycara *et al*., 1991). Like CYCLOPS, the CADET system was also an interactive system that left the adaptation and evaluation tasks to the designer. It provided a mechanical engineer with access to designs of simple mechanical devices, where each design case contained a causal model of the design. The goal was to enable the designer to do model-based adaptation of the past design.

Goel's KRITIK (Goel & Chandrasekaran, 1989; Goel *et al*., 1997) was the first autonomous case-based design system that addressed all the major subtasks of case-based design: retrieval, adaptation, evaluation and storage. KRITIK addressed the task of conceptual design in the domain of small, simple engineering devices. Given the specification of the function F, KRITIK generates a qualitative specification of the structure S. Since it addressed the F→S task, it stored an inverse S→B→F mapping in the form a structure–behavior–function (SBF) model in each past case. The SBF model provided the vocabulary for indexing the design cases as well as strategies for retrieving, adapting, verifying and storing design cases. JULIA (Hinrichs, 1992) was another early autonomous case-based design system. Although it operated in the informal domain of menu design, it addressed the important issue of changes in the design requirements during the design process.

3 Interactive design

The early nineties saw an explosion of interest in exploring CBR in interactive design. There were three major themes in this phase. The first theme was to annotate drawings of past designs not only with explanations of major design decisions but also with the outcomes of the designs (what worked, what did not and why), and the lessons learned from the outcomes. Kolodner and her colleagues built the ARCHIE series of interactive design aiding systems to explore this theme in the domain of building design (Domeshek & Kolodner, 1992; Pearce *et al*., 1992).

The second theme was to view design adaptation as a constraint satisfaction problem and hence to integrate CBR and constraint satisfaction. Smith, Faltings and their colleagues built the CADRE series of interactive design decision support systems that investigated this theme in the domain of building design (Hua *et al*., 1994; Dave *et al*., 1994). Their systems provided access to drawings of past designs and allowed the designer to make changes to these drawings. The system automatically propagated the constraints generated by the designer's decisions and alerted the designer of any conflicts.

The third theme was integration of case-based design support systems with CAD environments, which is important for building practical design support systems. Maher and her colleagues developed a series of interactive CAD systems (e.g. CADsyn, CaseCAD) that explored this theme in the domain of structural design (Maher & Zhang, 1993; Maher *et al*., 1995). Another important element of their work was the development of large repositories of design cases.

Perhaps the most ambitious effort in case-based interactive design was the FABEL system developed by Voß and her colleagues (Gebhardt *et al*., 1997). The FABEL system was also an interactive design decision support system for the domain of building design. It contained large, complex cases represented as annotated drawings, and used a range of techniques (e.g. rule-based reasoning, model-based reasoning) to help a designer in design adaptation. An especially noteworthy aspect of FABEL was its ability to retrieve drawings based on structural similarity with a query drawing.

4 Innovation in design

By the mid 1990s, research on CBR was beginning to seriously explore innovation and creativity in design. Faltings & Sun's (1996) interactive FAMING system investigated the design of a particular kind of mechanism embodied in antique wall clocks. An especially noteworthy aspect of FAMING (which means *invention* in Chinese) is its integration of functional reasoning with complex structural reasoning about shapes and spatial relations. FAMING perhaps was the first case-based system to strongly couple functional and spatial reasoning.

Bhatta developed the autonomous IDEAL system (Bhatta & Goel, 1997, 1998), which viewed the ability to transfer knowledge across distant domains as an important element of creativity. IDEAL abstracted teleological design patterns (e.g. open-loop feedback control) from design cases in one domain (e.g. operational amplifiers) and transferred them to design problems in another domain (e.g. gyroscopes). Its memory contained past design solutions at multiple abstraction levels ranging from design cases to design patterns, and it worked on design problems at different levels of abstraction.

Kolodner and her colleagues proposed an account of serendipity in creative design (Wills & Kolodner, 1994). In their account, a designer working on multiple design problems keeps different design goals in a working memory. As a partial solution to one problem becomes available (by internal problem solving or external input), the designer may connect the solution to a different design goal, thus addressing a different design problem. This account was later used to build a computational model of a historical case study about Alexander Graham Bell's invention of the telephone (Simina & Koldner, 1997).

Gómez de Silva Garza & Maher (1999) developed an evolutionary approach to design case adaptation. For design problems in which a function characterizing the fitness of the desired design is known, they represented design cases as genotypes, and used genetic algorithms to adapt past designs to invent a variety of new designs.

5 Other directions

All of the above theories and systems investigated CBR for preliminary phases of design in either architecture or engineering. However, there also has been some work that explored CBR in other domains of design and other phases of design. Smyth & Keane's (1996) Déjà Vu system was the first system to investigate case-based software design. The autonomous Déjà Vu system designed software for plant control. Two interesting aspects of Déjà Vu were its use of a hierarchically organized library of cases to enable CBR at multiple levels of abstraction, and retrieval of cases based on the needs of the adaptation task. Simpson and his colleagues developed AskJef (Barber *et al*., 1992), which was the first system to investigate case-based reasoning to explore the design of human-machine interfaces. AskJef, an interactive multimedia system, provided an HCI designer with libraries of past cases, where each case contained a series of annotated versions in the evolution of a design.

Chemical formulation is a design task where the final solution is a recipe consisting of an approximately constant number of components with relatively limited interactions between them. Therefore, this task is less open-ended than many design tasks because the components are better specified. Several case-based design systems have been developed for formulation tasks. The original example is General Electric Plastics' FormTool system, a colour matching system that is in everyday use (Cheetham & Graf, 1997; Cheetham, 2004). Further examples include pharmaceutical formulation for the design of tablets to deliver a specific dose of a drug (Craw *et al*., 1998; Wiratunga *et al*., 2002) and the formulation of rubber compounds for tyres (Bandini *et al*., 2004; Bandini & Manzoni, 2001), each of which highlights the need for adaptation knowledge to achieve a stable product that meets requirements.

As mentioned in the introduction, design involves many stages such as requirements analysis, preliminary design, geometric modeling, etc. Much of the research on case-based design as described above has focused on preliminary design (i.e. conceptual, qualitative design). However, some work has also explored later phases of design, and in particular, the phase of assembly and disassembly of designs. Purvis & Pu's (1995) COMPOSER was the first system to use CBR for assembly planning in which the task was to develop a sequence of actions for assembling a product from its components. The autonomous COMPOSER system operated in the domain of simple mechanical devices, using constraint satisfaction techniques to adapt past cases of assembly planning. Murdock's REM system (Murdock & Goel, 2001) autonomously adapted the software architecture of a disassembly planner to generate an assembly planner. REM thus performed meta-case-based reasoning in which the architecture of an agent that solves some task is adapted to address a related task.

6 Contributions to CBR

Two characteristics of the above research on CBR in design are especially noteworthy. Firstly, unlike much work on CBR, research on CBR in design has emphasized the importance of the adaptation task. This is in part because design cases can be quite complex and thus design adaptation is especially hard, and partly because of the difficulty of populating and organizing large case memories of complex designs. Secondly, research on CBR in design has not only applied and integrated CBR to design but also made significant theoretical contributions to CBR itself. To summarize, research on CBR in design thus far has made the following main contributions to CBR:

- grounded the computational process of CBR in content theories of domain models (e.g. KRITIK (Goel & Chandrasekaran, 1989; Goel *et al*., 1997));
- integrated CBR with a variety of other computational techniques ranging from model-based reasoning (e.g. FABEL (Gebhardt *et al*., 1997)) to constraint satisfaction problems (e.g. CADRE (Hua *et al*., 1994; Dave *et al*., 1994)) to genetic algorithms (e.g. Gómez de Silva Garza & Maher, 1999);

- integrated functional reasoning and spatial reasoning (e.g. FAMING (Faltings & Sun, 1996));
- used cases and CBR at multiple levels of abstraction (e.g. Déjà Vu (Smyth & Keane, 1996)); and
- provided new insights into innovation (e.g. IDEAL (Bhatta & Goel, 1997, 1998)).

These contributions have been driven by the eight challenging factors of design that were highlighted at the start of this paper. They demonstrate the advanced CBR techniques that have resulted from the specific challenges of the design task.

Acknowledgements

The authors would like to thank Mary Lou Maher for early discussions about the landmark papers in case-based design that are central to this article.

References

Alexander, C, 1964, *Notes on the Synthesis of Form*. Cambridge, MA: Harvard University Press.

Bandini, S, Colombo, E, Sartori, F and Vizzari, G, 2004, Case based reasoning and production process design: the case of P-truck curing. In *Proceedings of the Seventh European Conference on Case-Based Reasoning*. Berlin: Springer, pp. 504–517.

Bandini, S and Manzoni, S, 2001, CBR adaptation for chemical formulation. In *Proceedings of the Fourth International Conference on Case-Based Reasoning*, Vancouver, BC. Berlin: Springer, pp. 634–647.

Barber, J, Bhatta, S, Goel, A, Jacobson, M, Pearce, M, Penberthy, L, Shankar, M, Simpson, R and Stroulia, E, 1992, AskJef: integration of case-based and multimedia technologies for interface design support. In Gero, JS (ed.) *Artificial Intelligence in Design '92*. Dordrecht: Kluwer, pp. 457–474.

Bhatta, S and Goel, A, 1997, A functional theory of design patterns. In *Proceedings of the International Joint Conference on Artificial Intelligence, Nagoya, Japan*, pp. 294–300.

Bhatta, S and Goel, A, 1998, Learning generic mechanisms for innovative strategies in adaptive design. *Journal of Learning Sciences* **6**(4), 367–396.

Cheetham, W, 2004, Tenth Anniversary of the Plastics Color Formulation Tool. In *Proceedings of the Sixteenth Conference on Innovative Applications of Artificial Intelligence*. Menlo Park, CA: AAAI Press/The MIT Press, pp. 770–776.

Cheetham, W and Graf, J, 1997, Case-based reasoning in color matching. In *Proceedings of the Second International Conference on Case Based Reasoning, Providence, RI*. Berlin: Springer, pp. 1–12.

Craw, S, Wiratunga, N and Rowe, R, 1998, Case-based design for tablet formulation. In *Proceedings of the Fourth European Workshop on Case-Based Reasoning*. Berlin: Springer, pp. 358–369.

Dave, B, Schmitt, G, Faltings, B and Smith, I, 1994, Case based design in architecture. In Gero, JS and Sudweeks, F (eds) *Artificial Intelligence in Design '94*. Dordrecht: Kluwer, pp. 145–162.

Domeshek, E and Kolodner, JL, 1992, A case-based design aid for architecture. In Gero, J. S. (ed.), *Artificial Intelligence in Design '92*. Dordrecht: Kluwer.

Faltings, B and Sun, K, 1996, FAMING: supporting innovative mechanism shape design. *Computer-Aided Design* 28(3), 207–216.

Gebhardt, F, Voß, A, Gräther, W and Schmidt-Belz, B, 1997, *Reasoning with Complex Cases*. Norwell, MA: Kluwer.

Goel, A, Bhatta, S and Stroulia, E, 1997, KRITIK: an early case-based design system. In Maher, M and Pu, P (eds) *Issues and Applications of Case-Based Reasoning in Design*.Mahwah, NJ: Lawrence Erlbaum Associates, pp. 87–132.

Goel, A and Chandrasekaran, B, 1989, Functional representation of designs and redesign problem solving. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*. Los Altos, CA: Morgan Kaufmann, pp. 1388–1394.

Gómez de Silva Garza, A and Maher, ML, 1999, An evolutionary approach to case adaption. In *Proceedings of the Third International Conference on Case-Based Reasoning*. Berlin: Springer, pp. 162–172.

Hinrichs, T, 1992, *Problem Solving in Open Worlds: A Case Study in Design*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hua, K, Smith, IFC and Faltings, B, 1994, Integrated case-based building design. In *Proceedings of the First European Workshop on Case-Based Reasoning*. Berlin: Springer, pp. 436–445.

Maher, ML, Balachandran, B and Zhang, DM, 1995, *Case-based Reasoning in Design*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Maher, M and Pu, P (eds), 1997, *Issues and Applications of Case-Based Reasoning in Design*. Mahwah, NJ: Lawrence Erlbaum Associates.

Maher, ML and Zhang, DM, 1993, CADsyn: a case-based design process model. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **7**(2), 97–110.

Murdock, JW and Goel, AK, 2001, Meta-case-based reasoning: using functional models to adapt case-based agents. In *Proceedings of the Fourth International Conference on Case-Based Reasoning*. Berlin: Springer, pp. 407–421.

Navinchandra, D, 1991, *Exploration and Innovation in Design*. New York: Springer.

Pearce, M, Goel, A, Kolodner, J, Zimring, C, Sentosa, L and Billington, R, 1992, Case-based decision support: a case study in architectural design. *IEEE Expert* **7**(5), 14–20.

Purvis, L and Pu, P, 1995, Adaptation using constraint satisfaction techniques. In *Proceedings of the First International Conference on Case Based Reasoning*. Berlin: Springer, pp. 289–300.

Simina, M and Kolodner, JL, 1997, Creative design: reasoning and understanding. In *Proceedings of the Second International Conference on Case Based Reasoning*. Berlin: Springer, pp. 587–598.

Smyth, B and Keane, M, 1996, Design à la Déjà Vu: reducing the adaptation overhead. In Leake, DB (ed.) *Case-Based Reasoning: Experiences, Lessons and Future Directions*. Menlo Park, CA: AAAI Press/The MIT Press, pp. 151–166.

Sycara, K, Guttal, R, Koning, J, Narasimhan, S and Navinchandra, D, 1991, CADET: a case-based synthesis tool for engineering design. *International Journal of Expert Systems* **4**(2), 157–188.

Wills, LM and Kolodner, JL, 1994, Towards more creative case-based design systems. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press/The MIT Press, pp. 50–55.

Wiratunga, N, Craw, S and Rowe, R, 2002, Learning to adapt for case-based design. In *Proceedings of the Sixth European Conference on Case-Based Reasoning*. Berlin: Springer, pp. 423–437.