ELSEVIER

Knowledge-Based Systems 9 (1996) 79–81

Knowledge-Based SYSTEMS

Editorial

Reuse of designs

P.M. Wognum[a], I.F.C. Smith[b]

[a]*Department of Technology and Organisation, School of Management Studies, University of Twente, P.O. Box 217, 7500 AE Enschede, Netherlands*
[b]*LIA/EPFL, 1015 Lausanne, Switzerland*

1. Introduction

Computer support for design requires the acquisition, analysis and representation of large amounts of knowledge. Moreover, many different sources and types of knowledge are needed. The representation and use of knowledge in design systems requires a thorough understanding of the design process and design methods. However, there are no general methods yet which are available to synthetize complete designs from initial specifications. In addition, much knowledge is often informal and even unknown, and this is especially true for the conceptual phases of design. Human designers often reuse former experience in designing new systems. They recall similar designs and adapt them to requirements of the new situation. The reuse of such experience is also useful for enhancing the efficiency of design support systems, for providing explanations and for compensating for the lack of formal knowledge. A growing number of researchers are investigating models and techniques for reusing former solutions for design as well as many other applications.

The reuse of designs consists of two major aspects, which are also the major research themes in case-based reasoning. These aspects are retrieval and adaptation. Retrieval consists of retrieving old design cases.

Adaptation, also called 'redesign', or 'modification', consists of changing an old design to satisfy new requirements. Research in the field has been performed for several years and has led to a number of results. For example, CYCLOPS [1], KRITIK [2], CADET [3], ARCHIE [4], CADSYN [5], CADRE [6,7], JULIA [8,9], SEED [10], the FABEL project [11–13], BOGART [14], and ASA [15] are case-based design (CBD) systems proposed for mechanical engineering, civil engineering and architecture. However, despite the many interesting proposals, many problems need to be solved before applications become useful in practice. To support the retrieval of cases, for instance, suitable classifications have to be found for modeling old cases. The classification of cases is determined for a large part by the similarity defined between a new problem and the existing cases. Questions in this area are as follows:

- What features are relevant?
- What types of features need to be used?
- What is the relationship between the context of a new problem and the features to be used for retrieving a case?
- How specific or generic must features be?
- How complex are features?
- Which aspects need to be taken into account to assess similarity?

In the same vein, to be able to support adaptation, several questions need to be answered, for example the following:

- Which parts need to be modified?
- Is case combination required?
- Which methods can be used to adapt a design?
- How can we estimate the effects of the adaptation on the overall design?
- What aspects determine the best solution?

Below, we introduce some recent approaches to case reuse for design tasks. Application areas discussed are software design and engineering design in several technical areas. Although at a high level of abstraction software design is similar to engineering design, there are major differences between these two design areas. For example, in engineering design the spatial character must be considered from the outset of design, involving dealing with continuous variables. In software design, on the other hand, the behavior of the components to be used in the design is often difficult to estimate beforehand.

2. Approaches to reuse

We will discuss the approaches presented in this special issue of *Knowledge-Based Systems* by means of a 4-step model of reuse [16]. In this model the reuse of

0950-7051/96/$15.00 © 1996 Elsevier Science B.V. All rights reserved
SSDI 0950-7051(95)01020-3

designs is presented as a process consisting of the following four steps:

- Retrieval.
- Determination of the focus of adaptation. This usually involves *analysis* of the case in the new context.
- The generation of new specifications for the part to be adapted. Generation may be done using techniques such as *interpretation* by domain specific models and user interaction.
- Adaptation. This step involves finding a *feasible solution* for the case considering new specifications. When no feasible solution is available, control may revert to any one of the previous steps.

Each step will be discussed in a separate section.

2.1. Retrieval

In developing retrieval mechanisms for reusing old cases, we may distinguish between classification tasks and design tasks. In classification tasks the retrieved case is often used for explanation only, while in design the case is used as a first proposal for a solution to a new design problem.

A claim often made in favor of case-based reasoning is that the knowledge-acquisition bottleneck is avoided by reusing old cases. Maurer shows in his discussion of current research that this is only partly true. Even in retrieval, knowledge is needed to arrive at better and simpler indexing structures leading to an improvement of the performance of the retrieval process. A case study is presented which shows the benefits of knowledge modeling on retrieval. As a consequence, in practical applications, especially in complex domains, a tradeoff has to be made between the time spent on knowledge acquisition and knowledge modeling and the efficiency of the retrieval process.

In current practice, solutions are often expressed in a way that is very specific to the problem at hand. In other words the components used for the solution are 'functionally fixed'. As a consequence, it is difficult to see that existing components can be used to solve similar problems and to design reusable components. To allow and improve design with reuse it is also important to design for reuse, as argued by Dusink and Latour. In designing for and with reuse, assumptions must be challenged, implications must be made clear, analogies must be made explicit, and previous experience and domain knowledge must be emphasized. This means that the functional fixedness of components must be reduced, thus making the components more generally applicable. Reducing the functional fixedness of components is often related to a particular context. This context itself may be subject to functional fixedness, leading to an amplification of the problem.

A similar argument is presented by Pos et al. They discuss the use of a library of model fragments for modeling and simulation of engineering devices. To support the reusability of such model fragments, the assumptions underlying these fragments have to be made explicit. These assumptions concern both the content of the model fragments and the use of the fragments in particular contexts.

A third argument for spending more time on knowledge acquisition and modeling to improve the performance of case retrieval is presented by Smyth and Keane. They discuss a study on the use of adaptation knowledge to guide the retrieval of cases. It is shown that, by taking into account the possibilities for adaptation, the retrieval leads to the selection of the most adaptable cases. The additional costs spent during retrieval are greatly compensated for by the reduced costs of adaptation. In addition, the adaptation knowledge is represented explicitly, thus improving the flexibility of the retrieval system.

Finally, background knowledge may improve the structuring of cases in a case base. Brazier et al. present a compositional architecture of knowledge-based systems which support the modeling of existing solutions to design problems. Similarly, van Eldonk et al. and Pos et al. base the structure of the case base on a general ontology for structuring engineering design knowledge and an ontology for modeling simulation model fragments, respectively. The use of background knowledge provides us with explicit dependencies between parameters describing cases, which are often absent in ⟨feature,value⟩ descriptions as used in most case-based reasoning approaches.

2.2. Determine the focus of adaptation

Before we can start to adapt an old design to meet the new specifications, the parts of the design must be found which cause divergence from the new specifications. After that, a selection must be made of the part which can best be adapted. The selection may be based on many different criteria such as the ease of adaptation or the costs.

The parts which diverge from the new specifications are often found by comparing the ⟨attribute, value⟩ pairs of the old design and the new specifications (see the paper by Smyth and Keane). The selection of the part to be adapted is based on the adaptation knowledge related to the attributes of that part. Pos et al. determine which assumptions are violated in the old model by new observations which act as specifications for the new system to be modeled. The assumption violations lead to suggestions for adding or removing a model component. The changes to the model may lead to new violations of assumptions.

Brazier et al. determine which parts of requirement descriptions and design object descriptions are to be modified, and how, on the basis of explicit metaknowledge of design strategies (preferences, assumptions, etc). Both aspects may be determined by a user of the system.

van Eldonk et al. use techniques from model-based diagnosis to determine the cause of the discrepancy between the old design and the new specifications. This is possible because of the rich structure of the case base. The cause is a part that should be 'repaired', and it is called a diagnosis in model-based diagnosis terms. Many causes may be found, from which the one which it most suitable for adaptation can be selected.

2.3. Generate new specifications

When a part is found which should be adapted to allow the design to meet the new specifications, new partial specifications must be created to guide the adaptation of this part. Respecification, however, it often not needed because of the direct link between the new specifications and the description of the old design. In these situations, the specifications of the part to be adapted are a subset of the new specifications (see the papers by Brazier et al. and Smyth and Keane).

In the work of Pos et al. the new specifications of the parts to be added or removed are left implicit in the methods used to undo the assumption violations. van Eldonk et al. explicitly generate new specifications for the selected part since the relations between this part and the new specifications may not always be obvious. The generation of the new partial specifications is part of the determination of the diagnosis. The new partial specifications must be in accordance with the overall requirements. In addition, a restriction of the complexity of the adaptation to, for instance, parametric adaptation influences the diagnosis and specification task.

2.4. Adapt

Adaptation is difficult since the effects of changes on the overall design often play an important role. Although nearly any retrieved design requires adaptation, the importance of adaptation research has only recently been recognized. Research by Smyth and Keane has contributed to this increased awareness by showing that people choose adaptable cases and that adaptation-based retrieval improves performance. Some of this work is explained in this issue. In addition, it can be argued that, in the long term, problem specific modification of cases, or case adaptation and combination, will be recognized as *the* core research problem in case-based reasoning. Indeed, many issues related to case retrieval may soon be adopted by the much larger community that is concerned with information retrieval. Such appropriation is unlikely to happen with adaptation and combination research; these issues are unique to the study of case-based reasoning.

3. Conclusions

In spite of the important advances described in these papers, many questions remain. We hope that the results reported in this issue will stimulate many more interesting research efforts so that computer aided redesign will improve, on a large scale, the performance of designers working in practice. We would like to thank the following reviewers for their help in reviewing papers for this issue: J.M. Akkermans, R.R. Bakker, E.M. Dusink, J. Treur, I.F.C. Smith, F. Maurer and F.M.T. Brazier.

References

[1] D. Navinchandra, Case based reasoning in CYCLOPS, in *Proc. DARPA Case-Based Reasoning Workshop*, 1988, pp. 286–291.

[2] A.K. Goel and B. Chandrasekaran. Use of device models in adaptation of design cases, in *Proc. DARPA Case-Based Reasoning Workshop*, 1989, pp. 100–109.

[3] K.P. Sycara and D. Navinchandra. Influences: a thematic abstraction for creative use of multiple cases, in *Proc. DARPA Case-Based Reasoning Workshop*, 1991, pp. 133–144.

[4] A.K. Goel and J.L. Kolodner. Towards a case-based tool for aiding conceptual design problem solving, in *Proc. DARPA Case-Based Reasoning Workshop*, 1991, pp. 109–120.

[5] M.L. Maher and D.M. Zhang. Case-based reasoning in design, ion *Proc. Artificial Intelligence in Design*, Butterworth-Heinemann, 1991, pp. 137–150.

[6] K. Hua. Case-based design of geometric structures, *Thesis 1270*, Swiss Federal Institute of Technology, Lausanne, Switzerland, 1994.

[7] K. Hua, I. Smith and B. Faltings. *Integrated Case-Based Building Design*, Springer-Verlag, 1994, pp. 436–445.

[8] T.R. Hinrichs and J.L. Kolodner. The roles of adaptation in case-based design, in *Proc. DARPA Case-Based Reasoning Workshop*, 1991, pp. 121–132.

[9] T.R. Hinrichs, *Problem Solving in Open Worlds*, Lawrence Erlbaum, 1992.

[10] U. Fleming. Case-based design in the SEED system, *1st Computing Congress*, American Society of Civil Engineers, USA, 1994.

[11] S. Bakhtari, K. Börner, B. Bartsch-Spörl, C.-H. Coulon, D. Janetzko, M. Knauff, L. Hovestadt and C. Schlieder. EWCBR93: contributions of FABEL, *Fabel Report 17*, GSD, Germany, 1993.

[12] S. Bahktari and B. Bartsch-Spörl. Our perspective on using CBR in design problem solving, *1st European Workshop on Case-Based Reasoning*, Kaiserslautern, Germany, 1993.

[13] A. Voss. The need for knowledge acquisition in case-based reasoning — some experiences from an architectural domain, in *Proc. 11th European Conference on Artificial Intelligence*, John Wiley, 1994, pp. 463–467.

[14] J. Mostow, M. Barley and T. Weinrich. Automated reuse of design plans in BOGART, in C. Tong and D. Sriram (eds.), *Artificial Intelligence in Engineering Design*, Academic Press, 1992, pp. 57–103.

[15] A. Giretti, L. Spalazzi and M. Lemma. A.S.A. an interactive assistant to architectural design, in *Proc. Artificial Intelligence in Design '94*, Kluwer, 1994, pp. 93–108.

[16] R.R. Bakker, S.J.M. van Eldonk, P.M. Wognum and N.J.I. Mars. The use of model-based diagnosis in redesign, in A. Cohn (ed.), *ECAI94: 11th European Conference on Artificial Intelligence*, John Wiley, 1994, pp. 647–651.