this paper was originally published as:
Borner, K. (1993). Structural Similarity as guidance in case-based design. In Lecture Notes In Computer Science; Selected papers from the First European Workshop on Topics in Case-Based Reasoning. Vol. 837, pp. 197-208

Structural Similarity as Guidance in Case-Based Design *

Katy Börner

HTWK Leipzig, Dept. of Informatics,
P.O.Box 66, 04251 Leipzig, FRG,
katy@informatik.th-leipzig.de

Abstract. This paper presents a novel approach to determine structural similarity as guidance for adaptation in case-based reasoning (CBR). We advance structural similarity assessment which provides not only a single numeric value but the most specific structure two cases have in common, inclusive of the modification rules needed to obtain this structure from the two cases. Our approach treats retrieval, matching and adaptation as a group of dependent processes. This guarantees the retrieval and matching of not only similar but adaptable cases. Both together enlarge the overall problem solving performance of CBR and the explainability of case selection and adaptation considerably. Although our approach is more theoretical in nature and not restricted to a specific domain, we will give an example taken from the domain of industrial building design. Additionally, we will sketch two prototypical implementations of this approach.

1 Introduction

The effectiveness of case-based reasoning depends on the ability to determine former experiences (cases) that are useful and applicable to solve new, similar problems. When one tries to handle *synthesis* tasks as opposed to *analysis* tasks, however, the determination of similarity alone is not enough: The *adaptability* of former cases to problems of current interest becomes essential.

Most approaches to similarity assessment in CBR (c.f. [Kol92, Ric92]) estimate the usefulness of cases based on the presence or absence of certain features. The features are preclassified as important with respect to retrieval. Similarity is assessed by a numeric computation and results in a single number which is meant to reflect all aspects of the similarity. There are approaches (e.g., [Hin92]) which try to capture the plausible inferences intrinsically through the organization or *indexing* of knowledge. Constraints on a problem serve as indices into a design memory. The memory returns cases that provide a solution, some of the

* This research was supported by the German Ministry for Research and Technology (BMFT) within the joint project FABEL under contract no. 413-4001-01IW104. Project partners in FABEL are German National Research Center of Computer Science (GMD), Sankt Augustin, BSR Consulting GmbH, München, Technical University of Dresden, HTWK Leipzig, University of Freiburg, and University of Karlsruhe.

context of the design as well as feedback or external evaluation. This information is used to determine how applicable the case is, how to adapt it, and how to avoid repeating previous failures. Contrary to this, we want to introduce an approach to similarity assessment which depends not only on prior cases but also on the domain background knowledge available for adaptation. Structural similarity is defined as the most common structure of cases and the rules needed to determine it. It is able to guide solution adaptation.

Conventional CBR systems which use causal models (e.g., CASEY [Kot88], SWALE [KLO86], CHEF [Ham86]) treat retrieval, matching (or justification), and adaptation separately one after each other. By contrast, we will show that the integration of these stages improves the suitability and performance of CBR significantly.

The paper is organized as follows: First, we introduce a novel definition and use of structural similarity in CBR and its motivation by prior work. Second, to exemplify our approach, we solve a specific synthesis task taken from the domain of building design. Third, we sketch two implementations of this methodology. We conclude by discussing our approach and delineate future work.

2 Our Approach: Structural Similarity as Guidance

To introduce our general approach to similarity assessment and adaptation, we use Fig. 1. As a basis to assess similarity in terms of adaptability a canonical system of cases and associated case modification rules[2] is needed. Prior cases have to be represented both in attribute-based and structural forms (e.g., by terms or graphs) and are stored in the case-base on the right side of Fig. 1. Background knowledge is represented by domain-dependent and task-dependent rules (e.g., term or graph modifications like generalizations, geometrical transformations, and substitutions) including their 'inverse' rules. This knowledge is specific for classes of cases and will be stored separately in the rule-base shown in the middle of the figure. Changes in case or background knowledge influence both similarity assessment and adaptation.

Structural similarity assessment is a computationally expensive process. Inspired by the work of DEDRE GENTNER and KENNETH D. FORBUS [GF91], we will tune computationally cheap, fast preselection and expensive structural similarity assessment to complete each other. The procedure (marked by arrows) is as follows.

Given the new problem in attribute-based description, we start by determining a set of candidate cases, the surface attributes of which are similar to those of the new problem. Based on this computationally cheap analysis (*surface similarity assessment*) of the problem, we can now use the transformation function ϕ to translate the new problem into a structural representation. Corresponding to the new problem and the preselected candidate cases, modification rules f_1, f_2 will be chosen (*rule selection*) and applied until a common structure of the actual

[2] The rules used here are not covered by the expression rewriting rules.

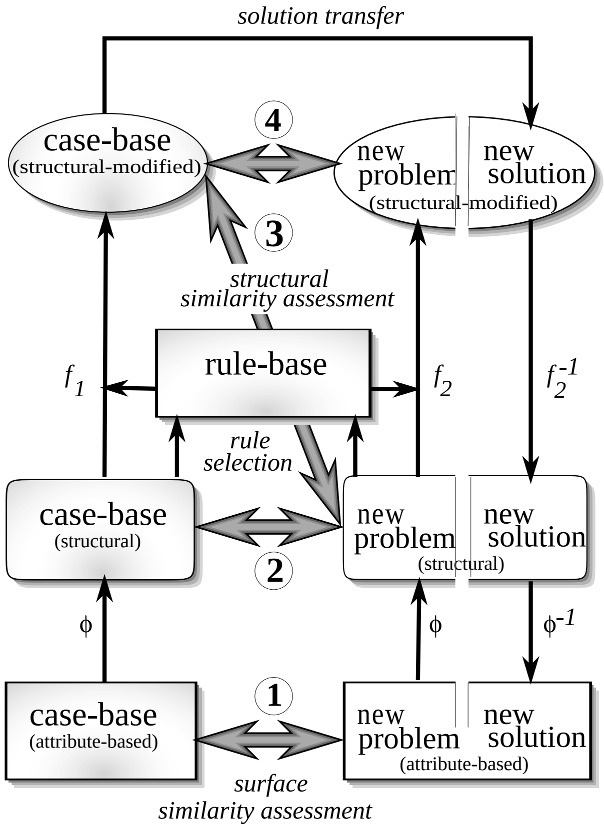

Fig. 1. General Approach and Different Levels to Assess Similarity

problem and one candidate case is found (*structural similarity assessment*). Note that $f_1 \neq f_2$ is possible. Now the solution of this candidate case, also modified, can be transferred to the new problem (*solution transfer*). After that, inverse modification rules f_2^{-1} are applied to get the concrete structural representation of the new solution. Using ϕ^{-1} we will get the attribute-based representation of the new solution[3]. This will be offered pictorially to the user.

The common structure of cases together with the modification rules applied to obtain them determine which prior solutions are useful. The inverse modification rules will show how to adapt them.

We should mention that similarity assessed by the system can considerably differ from that assessed by human users, even assuming that both have the

[3] In mathematical terms, it is not always possible to invert the mapping ϕ.Thus ϕ^{-1} denotes any computable flattening that takes a structural case description and transforms it into an attribute-based one such that applying ϕ would yield the structured representation, again. The same applies to f and f^{-1}.

same set of cases at hand. Due to the background knowledge available their strategies to use and reason from these cases can be quite different. In each case the adapted solution should meet the given requirements.

Before giving a practical example we want to contrast our approach to prior work. In addition Fig. 1 presents conventional approaches to similarity assessment in CBR and analogy (denoted by fat grey double arrows and enumerated (1) to (4)):

(1) Frequently used in CBR are *surface similarity assessments* based on attribute-based case representations (c.f. [Weß91, Ric92, Kol92]). They provide a single numeric value which is meant to reflect all aspects of similarity. If similarity is defined depending on some given distance it inherits the properties of reflexivity, symmetry, and triangular inequality. This is sometimes not desirable. An excellent discussion about this topic provides [Jan93].

(2) If interdependencies of attributes have to be taken into account, representations like terms and graphs are used as a basis for similarity assessments. Again, the output is a single numeric value.

(3) There are approaches where cases stored in the case-base will be modified (e.g., using letter substitution rules as known in speech recognition) to determine similarity. Similarity is defined by the transformations used to transform one case into another.

(4) The principle of redescription [Ind91, O'H92] modifies new and old problem descriptions in a mutually dependent way in order to synthesize an isomorphism of both descriptions.

We emphasize that all the approaches mentioned above define similarity independent on the adaptation knowledge available.

3 An Example: Case-Based Industrial Building Design

Much work has been done in case-based building design (c.f. [Goe89, DK92, Nav92, HF93]), which is one of the most complex real world synthesis tasks. Architects experience the world in cases. CBR seems to be the natural problem solving method. Synthesis tasks in building design usually aim at the creation of objects and their conflict free arrangement corresponding to certain requirements.

In our project, we focus on the installation of supply system nets in industrial buildings with a complex infrastructure. Fig. 2 illustrates the layout of subsystems for return air. Return air accesses have to be connected by return air connections. The ellipses used in the figure provide information on a sketch level of design. To use ellipses instead of rectangles is a very useful graphical trick: Ellipses overlap only in a few points. Thus, information on different levels of abstraction can be displayed simultaneously. For a detailed introduction into this representation scheme, the reader is referred to [Hov93]. In Fig. 2, thinly drawn circles denote places where accesses can be placed. Bold drawn circles

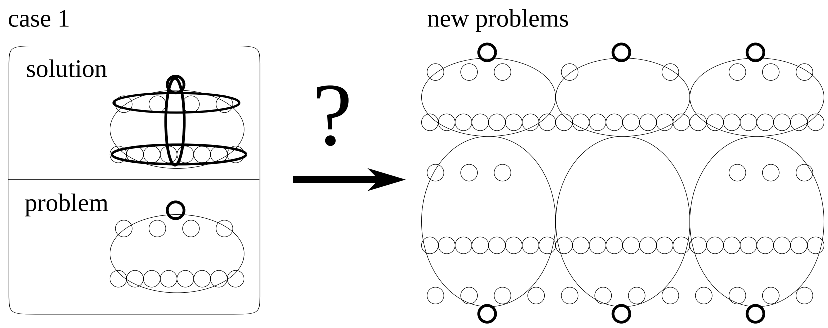

Fig. 2. Case-Based Design of Air Supply Nets

denote places reserved for main accesses. Ellipses denote areas where connections of supply accesses can be placed. The task to be tackled is the design of the connections for supplies that properly cover all of a given set of accesses for supply. Simply speaking, circles have to be covered by ellipses. The question is, how to transfer the experience stored in case 1 to the new problems at hand ? Identical problems occur seldom in design. Adaptation is essential. Using this task of connecting supply accesses, we will show how our approach works.

3.1 An Example: Knowledge Representation

The selection of an appropriate knowledge representation is strongly task and domain dependent. To realize our approach we need two different types of case representations as well as a complementing rule-based representation of proper case modifications.

The *attribute-based* case representation provides visually prominent features of objects. Following the work of Ludger Hovestadt [Hov93] each object (circle or ellipse) will be represented by its spatial dimensions and nine further attributes like *time* at which this object was created, *aspect* which assumes one of 'return air,' 'fresh air,' etc., and *morphology* which refers to 'access,' 'connection', etc. This fixed set of dimensions will be used as indices[4]. This representation permits us to use computer drawings as the main basis for man-machine interaction in building design. Architects can mark problems by simply manipulating objects in drawings. Solutions will be offered as drawings, too. Filling in large forms to represent each case or to describe a new problem will not work in real world applications.

Next, we have to encode case-based *structural* knowledge about relative positions and spatial arrangements of objects in a machine–usable form. Influenced

[4] To get cases in a less redundant form, 'space-coordinates' are normalized. Therefore the smallest x-, y-, and z-dimensions of each case are set to zero.

by the work of BIPIN INDURKHYA (c.f. [Ind91]), we represent the complex structures like supply air net structures as terms over some appropriately tailored signature. Therefore, a finite, heterogeneous, and finitary signature is assumed. This is taken as a basis for building terms and formulae, as usual. The detailed formal description of the signature used to represent cases structurally can be found in [Bör93]. Equational knowledge about functions and their relations is formalized . Note that each solution description contains the corresponding problem description. There is a function ϕ with its inverse which realizes the transformation of the attribute-based descriptions into structural ones and vice versa.

Additionally, we need knowledge (*rules*) about proper modifications of structural case representations. Terms can be modified using generalizations. To express generalized terms we need a sorted family of variables. For simplicity, we assume all variables to be called x, with indices whenever necessary. There are meaningful geometrical transformations like *reflection, rotation, translation*, etc. in our domain[5]. Additionally, structural representations can be modified using abstraction rules, which assign term expressions to constants (abstract attributes) like *row, regular, covered* etc. These three different kinds of modification rules including their inverses will be stored in the rule-base.

This knowledge provides a canonical system and the potential for structural similarity assessment which guides adaptation.

3.2 An Example: Similarity Assessment

For illustration, the main procedure given in Fig. 1 is exemplified in Fig. 3. The left, lowest box shows the pictorial and attribute-based representation[6] of one typical case stored in the case-base. By taking the functions *cover* and *copy* and object constants, combined with appropriate parentheses and commas, we are able to express the solution of this case structurally by *cover(copy(Y,3,Circle))*. This term stands for *take one circle, copy it three times and arrange them all in the y-direction. Afterwards cover all circles with a single ellipse.* The right, lowest box shows the pictorial and attribute-based representation of the new problem to be solved. Given in the same box but not available at this time is the solution of the new problem. The particular intention (also called subgoal) the user wants to concentrate on is the connection of air supplies.

The first initial analysis of the new problem can be done by an inexpensive *surface similarity assessment*, based on the attribute-based descriptions. The result is a set of candidate cases which have similar surface attributes such as aspect, number of objects etc. In this way, the rather large set of cases stored in the case-base can be reduced to a few useful candidate cases.

[5] In contrast to, e.g., mechanical engineering where functional dependencies constitute the background knowledge model, in building design topological dependencies play the paramount role and have to be considered during reasoning. Additionally, geometrical laws (reflecting a reflected figure again will result in the original figure) can be exploited.

[6] For simplicity, we only gave the values of the attributes x, dx, y, dy, time, aspect, and morphology of each object.

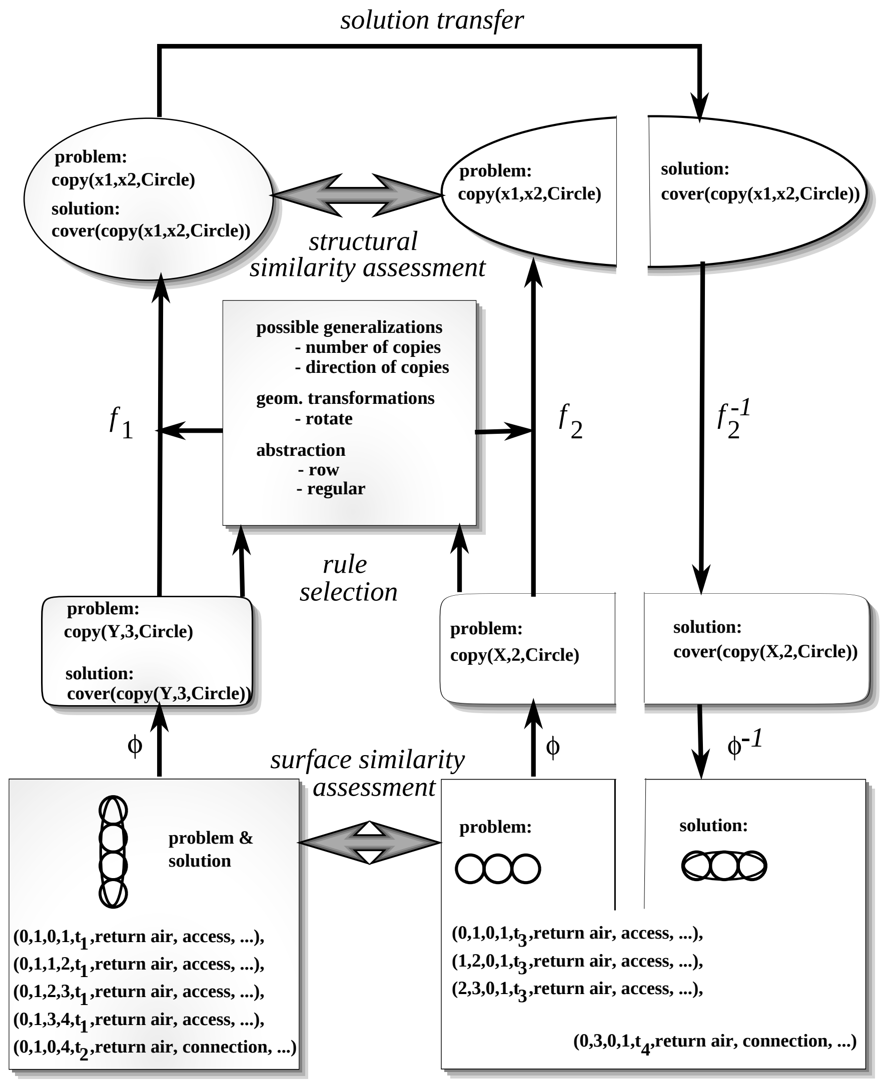

Fig. 3. An Example: Similarity Assessment and Adaptation

The next step is the transformation of the new problem into a structural representation. Here, candidate cases provide information about proper transformations referred to as ϕ. Thus, the new problem, which consists of three circles arranged in a row in the x-direction will be structurally represented by *copy(X,2,Circle)*.

Based on the structural representation of superficially similar prior cases, the more expensive *structural similarity assessment* is performed. Axioms and modification rules will be applied to determine the main structure which the new and a prior problem have in common. In our example, there are at least three different ways to achieve this:

- The first way uses generalization. For example, the concrete arrangement direction and the number of copies will be replaced by variables x_1 and x_2. The resulting common problem description will be *copy(*x_1, x_2*, Circle)* as shown in Fig. 3.
- The second way is to use generalization and geometrical transformation. Here, the number of copies will be generalized, too. One term representation will be rotated about 90 degrees.
- The third way uses abstraction. Here descriptions like *row* and *regular* will be used as abstract attributes. The idea behind this is that the more identical abstract attributes structural descriptions share, the more similar they are.

Given the main structure of both problem descriptions, we can simply transfer the also modified prior solution *cover(copy(*x_1, x_2*, Circle))* to the actual problem (in Fig. 3 referred to as *solution transfer*).

3.3 An Example: Adaptation

Using our knowledge about the sequence of modifications to determine the common structure, the transferred solution can be adapted to the new problem. This is denoted by f^{-1}. To get the concrete structural solution, in the example,

- where two generalizations were used to determine structural similarity, one has to replace x_1 by its former value X and analogously x_2 by 2. The resulting term will be *cover(copy(X,2,Circle))*.
- where generalization and geometrical transformation were used, one rotates the figure about -90 degrees (or 270 degrees) and replaces the variable number of copies by *2*. The resulting term will be *cover(copy(X,2,Circle))*.
- where abstraction was used, the transferred solution can be expressed by the attribute *covered*. The reverse concretization is somewhat difficult. Given terms and their corresponding abstract descriptions, one can try to find one structural representation which fulfils all abstract attributes (in this example *row, regular*, and *covered*). This suffices, if the number of these term-attribute assignments remains small but becomes intractable otherwise.

Given the structural representation of the new solution the application of the inverse transformation ϕ^{-1} yields the attribute-based and hence pictorial representation of the new solution.

4 Implementation

Our approach is prototypically implemented in SynTerm (for **Syn**thesis by using **Term**-based knowledge representations) and **SynGraph** (for **Syn**thesis by using

Graph-based representations). Both are integrated in a knowledge based system which operates in the domain of industrial building design. Whereas the overall system provides user interfaces and a broader range of functionality, SynTerm and SynGraph supply the system with solutions for special design problems. Thus SynTerm and SynGraph exchange solutions for problems of the system using the approach to similarity assessment and adaptation introduced. Fig. 4 illustrates the architecture of SynTerm and SynGraph. In the diagram, oval shapes represent data structures. Rectangular shapes represent processes.

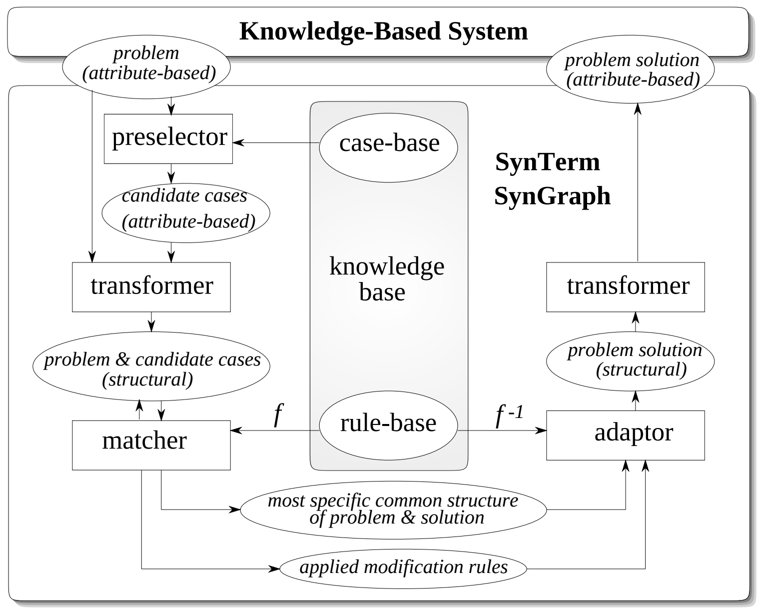

Fig. 4. Architecture of SynTerm and SynGraph

The CBR modules consist of a *preselector*, *transformers* (to transform the attribute-based representation into a structural one and vice versa), a *matcher*, and an *adaptor*. The knowledge base provides a canonical system of cases and rules. The input of SynTerm and SynGraph is an attribute-based problem description. Output is the corresponding solution in attribute-based representation. The first task is performed by the *preselector*. It defines a set of candidate cases from the case-base together with a suitable transformation function needed by the *transformer* to produce the structural problem description. The *matcher* applies rules to the current problem and the candidate cases until a most specific common structure of both can be determined. This most specific common structure of problems and the solution together with the applied modification rules are the input for the *adaptor*. In addition, the *adaptor* has access to the inverse modification rules stored in the rule-base. Output of the adaptor is the full specified problem solution in structural representation. The *transformer* provides its attribute-based representation.

Learning, e.g., the storage of new cases, new rules and the application of techniques like Knuth–Bendix completion to guarantee completeness of structural knowledge or learning of modification rule conditions or weights are not addressed in this implementation. This is one direction for future work.

The initial experimental results support the claims that the proposed approach offers improved adaptation facilities in synthesis tasks together with a greater overall explanation performance.

5 Conclusions

We presented a novel approach to similarity assessment and adaptation in Cbr. Structural similarity is assessed if and only if there is knowledge available to adapt old solutions correctly to the current problem of interest. Background knowledge is organized in a manner that permits the efficient identification of the appropriate cases and modification rules. The result is an approach to structural similarity which implies adaptability. Retrieved cases are adaptable cases. Only one similar, adaptable case is needed and looked for.

The integration of retrieval, matching and adaptation essentially improves case-based problem solving, especially for synthesis tasks where adaptation is important. In addition, the structural similarity assessment provides a basis for more descriptive explanations for why particular solutions have been selected and adapted.

The basic mechanisms of our approach are domain independent and thus facilitate the adoption of the technique across a range of Cbr application domains like software design, machinery, and technical configuration. Detailed investigations are under work.

6 Acknowledgements

This research has been strongly inspired by work done in the project Fabel the general objective of which is the integration of case-based and model-based approaches in knowledge-based systems. This work was guided by Bipin Indurkhya's work on analogical reasoning. He contributed to this work by providing detailed and valuable comments. I wish to thank Dietmar Janetzko for his helpful comments on an earlier draft. I am indebted to my thesis advisor Klaus P. Jantke for his continuing support and encouraging feedback. Phil Watson deserves thanks for his aid in revising an earlier version of this paper. Nonetheless, the paper reflects our personal view.

References

[Bör93] Katy Börner. Structural similarity in case-based design. In Dietmar Janetzko and Thomas J. Schult, editors, *Fälle in hybriden Systemen. Beiträge zum 2. Workshop des Arbeitskreises fallbasiertes Schließen*, pages 1–13, 1993.

[DK92] Eric A. Domeshek and Janet L. Kolodner. A case-based design aid for architecture. In *Proc. Second International Conference on Artificial Intelligence in Design*, pages 497–516. Kluwer Academic Publishers, June 1992.

[GF91] Dedre Gentner and Kenneth D. Forbus. MAC/FAC: A model of similarity-based retrieval. In *Proceedings of the Cognitive Science Conference*, pages 504–509, 1991.

[Goe89] Ashok K. Goel. *Integration of case-based reasoning and model-based reasoning for adaptive design problem solving.* PhD thesis, Ohio State University, 1989.

[Ham86] Kristian J. Hammond. CHEF: A Model of Case-Based Planning. In *Proceedings of the 5th Annual National Conference on Artificial Intelligence (AAAI-86)*, pages 267–271. Morgan Kaufmann Publishers, 1986.

[HF93] Kefeng Hua and Boi Faltings. Exploring case-based building design – CADRE. *AI EDAM*, 7(2):135–143, 1993.

[Hin92] Thomas R. Hinrichs. *Problem solving in open worlds: A case study in design.* Lawrence Erlbaum Associates, 1992.

[Hov93] Ludger Hovestadt. A4 - digital building - extensive computer support for design, construction, and management of buildings. In *Proc. CAAD Futures'93, Pittsburgh, USA*, page to appear, 1993.

[Ind91] Bipin Indurkhya. On the role of interpretive analogy in learning. *New Generation Computing*, 8(4):385–402, 1991.

[Jan93] Klaus P. Jantke. Nonstandard concepts of similarity in case-based reasoning. In *Proceedings of the 17th Annual Meeting of the German Society for Classification in Kaiserslautern.* Springer Verlag, 1993.

[KLO86] Alex M. Kass, David B. Leake, and Christopher C. Owens. SWALE: A program that explains. In Roger C. Schank, editor, *Explanation patterns: Understanding mechanically and creatively*, pages 232–254. Lawrence Erlbaum Associates, 1986.

[Kol92] Janet L. Kolodner. An introduction to case-based reasoning. *Artificial Intelligence Review*, 6:3–34, 1992.

[Kot88] Phyllis Koton. *Using experience in learning and problem solving.* PhD thesis, MIT, 1988.

[Nav92] D. Navinchandra. Innovative design systems, where are we and where do we go from here? Part I: Design by association. *The Knowledge Engineering Review*, 7:3:183–213, 1992.

[O'H92] Scott E. O'Hara. A model of the 'redescription' process in the context of geometric proportional analogy problems. In Klaus P. Jantke, editor, *Proceedings of the International Workshop on Analogical and Inductive Inference*, pages 268–293. Springer Verlag, 1992.

[Ric92] Michael M. Richter. Classification and learning of similarity measures. In *Proceedings of the 16th Annual Meeting of the German Society for Classification in Kaiserslautern.* Springer Verlag, 1992.

[Weß91] Stefan Weß. PATDEX/2: Ein System zum adaptiven, fallfokussierenden Lernen in technischen Diagnosesituationen. SEKI-Working Paper SWP91/01, University of Kaiserslautern, Germany, 1991.