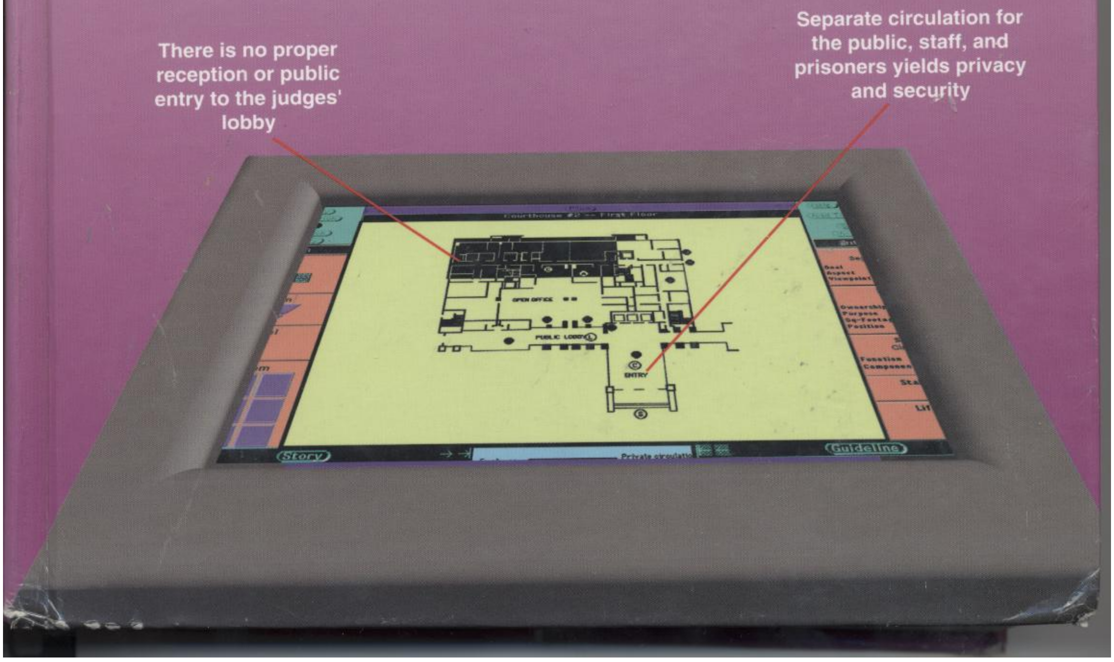
CASE-BASED REASONING

Janet Kolodner



Publisher and Editor *Michael B. Morgan*
Assitant Editor and Permissions *Douglas Sery*
Project Management *Professional Book Center*
Design, Composition, Figure Rendering *Professional Book Center*
Copyediting *Virginia Rich*
Cover Design *Terry Earlywine*

Library of Congress **Cataloging-in-Publication Data**
Kolodner, Janet L.
Case-based reasoning / Janet Kolodner.
p. cm.
Includes bibliographical references and index.
ISBN 1-55860-237-2
1. Expert systems (Computer science) I. Title.
QA76.76.E95K64 1993
006.3'3--dc20 93-35703
CIP

Printed in the United States of America
97 96 95 94 93 5 4 3 2 1

Morgan Kaufmann Publishers, Inc.
Editorial Offices
2929 Campus Drive, Suite 260
San Mateo, CA 94403

Contents

PART III Retrieving Cases from the Case Library 283

Preface

Every working professional I know wants a wife, the old-fashioned kind—one who will wait on him or her, do the household tasks, and in general, make life easier. I want one of these too. I call it the super-housewife robot.

There are lots of things I want my robot to do: plan and make meals; do the housecleaning, laundry, dishwashing, and chauffering; keep me informed of things on my schedule that are easily forgotten; pay the bills; warn me of problems that might arise or anything abnormal it discovers while performing its duties. I want it to learn from its mistakes; I won't like it if mistakes are repeated. I'd like it to be able to manage its time well. It ought to get better at scheduling activities over time, finding out about pitfalls and time sinks and not falling into them. In general, I want it to do the things that will allow me to have more leisure time, and I want it to do those things well enough so that I won't be tempted to do them myself.

My robot is going to have to do many of the same things I do from day to day. It will have to be a good planner and problem solver, it will have to both reason and act, and it will have to learn from its experiences how to do things in the way I want them done. And it will have to work in real time—no time to figure everything out from scratch all the time.

Over the past several years, several projects in my lab have been aimed at creating this super-housewife robot. JULIA is a meal planner. EXPEDITOR schedules household tasks. MEDIC worries about combining planning and execution. MEDIATOR mediates children's disputes. The projects, of course, have much broader applicability than my robot. JULIA is the prototype for a design problem solver, and its heuristics could just as well be used for architectural or mechanical design as for meal planning. EXPEDITOR could be used to schedule army maneuvers just as well as it could be used to schedule household tasks. MEDIC was designed to be a doctor and is now being tested in an underwater robotics domain. And MEDIATOR can reason about world crises as well as about children's disputes.

The super-housewife robot, then, is a combination of all the reasoners we hope to build someday. How can we build these reasoners that can design, plan, schedule, negotiate, integrate planning with activity, and learn and do it all in real time? We might be tempted to look at all these tasks separately and solve each of them alone, but that would probably be a mistake. I keep the super-housewife robot in mind because I want to solve these problems in a parsimonious way. I want the solutions to all these problems to be sufficiently similar that we can put them into one cognitive architecture and make them run in conjunction with one another. And I want learning to be integrated with reasoning, not to be built on top.

I'm not the first to dream the dream of the ultimate intelligent machine, nor am I the first to propose a way to build it. I am proposing *case-based reasoning* as a means of getting there. In case-based reasoning, new problems are approached by remembering old similar ones and moving forward from there. Situations are interpreted by comparing and contrasting them with previous similar situations. Stories are understood and inferences are made by finding the closest cases in memory, comparing and contrasting with those, making inferences based on those comparisons, and asking questions when inferences can't be made. And learning happens as part of the process of integrating a new case into memory.

This approach is appealing for a variety of reasons. First, the process is relatively simple. It allows a reasoner to copy what has been done before even if the reasoner doesn't understand what is going on. We all perform mindlessly from time to time. With all the demands on our time and thoughts, we need to. And so might the super-housewife robot or the Mars rover or any other intelligent machine. Case-based reasoning allows a reasoner to solve problems with a minimum of effort. Second, case-based reasoning provides a way of dealing with an uncertain world. If we can't predict what might happen with certainty, or if we are missing knowledge we would like to have, we depend on the world's being continuous. What was true yesterday is likely to be true today. Cases record the past, giving us and our computers a way to make assumptions about the present. Third, the process seems intuitively plausible. It seems like what we, as people, do quite often. This plausibility has several implications. It might be easier to capture an expert's knowledge in the form of cases than in the form of rules, so building expert systems might be easier in a case-based paradigm. And, if we understand human reasoning well enough to mimic it in a computer, then we might use the results not only to build automated systems but also to build interactive systems that interact with people in a natural way.

Though we are still far from building the super-housewife robot, I believe case-based reasoning provides the basis for a cognitive model that will allow us to build it someday and, in the meantime, will allow us to build a variety of automated and interactive systems that can help us with our tasks in the workplace, in the home, and even (dare I say it?) on the battlefield (though I would prefer that we solve our problems by more peaceful means—perhaps case-based negotiators can help with that). This book presents the state of the art in case-based reasoning. It provides our current answers to many questions: how to represent knowledge in cases, how to index cases for accessibility, how to implement retrieval processes for efficiency, how to adapt old solutions to fit new situations, and so on. It is also honest about the open problems.

In writing this book, I've tried to address two audiences: the cognitive science and artificial intelligence research communities, who want to find out more about the ins and outs of case-based reasoning as it relates to research they are doing, and members of the expert-systems building world, who want guidelines for building working systems. Sometimes it was easy to address both; other times hard. A section at the end of chapter 1 gives some guidelines for reading the book, different ones for each of the two communities. In trying to address both, I found that the book grew and grew, and that in the end, I hadn't addressed some issues as thoroughly as I had wanted to. The cognitive model behind case-based reasoning is discussed at length, for example, but recent experimentation testing the psychological implications of work done in the case-based reasoning community is missing. Methodologies for addressing system-building issues are covered, but code and pseudocode for implementing these approaches is missing. The components of a case-based system are covered in detail, but the learning that emerges from the system as a whole is only covered briefly.

In the end, I was afraid that maybe I had not tied the pieces together well enough. But the book was long, and I didn't think readers would want to read another chapter (I certainly didn't want to write one). The solution is the case library of case-based reasoning systems in the Appendix. I sent e-mail to people I knew who had written case-based systems, and they passed my message on to others. I asked for descriptions of their systems and the lessons they had learned. The result is quite exciting—I didn't know there were so many case-based systems out there, and I didn't know there were so many fielded ones. And many of the lessons learned are informative not only for those just getting started but also for those of us who are experts.

I learned a variety of lessons myself from the replies. For example, shallow indexing of a flat memory, similar to an inverted index, is by far the most common means of indexing case-based systems—I wouldn't have guessed that. I added more text to the book about the ins and outs of such a scheme after I learned that. I also wouldn't have guessed that the nearest-neighbor approach to matching and ranking cases was so extensively used or that there were so many variations on choosing a best case. Nor did I know how large some case libraries are—one is over twenty thousand cases, and many others have thousands of cases. Surprisingly, those systems do retrieval efficiently using the simplest retrieval methods. I was also surprised, though I shouldn't have been, by the huge variations in sizes of cases. Some cases are only a few attributes big, others are several hundred kilobytes large. I suspect that the largest case libraries that use simple methods for retrieval work well on small cases but would be hard to scale up to large cases, but I don't know. There are also several lessons about the representational form of cases. One lesson is that though case representations should be fully expressive, they (or their indexes) need to be structured in flat rather than structured form for easy matching. And many people reported on their use of *snippets*, parts of cases of varying sizes, in solving problems. In addition, I learned some lessons about some things that perhaps should have been covered in the book—things that were so obvious to me that they didn't make it into the book but obviously not so obvious to system builders. That's not to belittle system builders; rather, it is a statement about how little teachers often know about what their students need to learn. In any case, the Appendix does tie the pieces of a case-based system together, and I highly recommend reading it. My thanks to all of you who replied to my request for information.

For those who want more information than is in this book, let me recommend some additional reading. First, Schank and Abelson's (1977) *Scripts, Plans, Goals, and Understanding* and Schank's (1982) *Dynamic Memory* provide the history behind the creation of case-based reasoning. Case-based reasoning was conceived in the late 1970s while I was a graduate student at Yale (Roger Schank was my advisor), and the first systems began to be built in the early 1980s, by my students and Roger's. At the same time, members of the AI legal reasoning community, most notably Edwina Rissland, were looking at the role of cases in argumentation. As the world began to be populated with people working on case-based reasoning, several books were published that capture snapshots of the case-based reasoning world. *Experience, Memory, and Reasoning*, by me and Chris Riesbeck (1986), holds a snapshot of 1984 case-based reasoning capabilities. The 1988, 1989, and 1991 Defense Advanced Research Projects Agency (DARPA) Case-Based Reasoning Workshop proceedings (Kolodner 1988b; Hammond 1989c; Bareiss 1991) hold snapshots of case-based reasoning research and development in those years. Riesbeck and Schank's (1989) *Inside Case-Based Reasoning* provides a snapshot of work in case-based reasoning at Yale. More recently, the most influential of the Ph.D. theses on case-based reasoning have been published as books (Hammond 1989a; Bareiss 1989a; Ashley 1990; Hinrichs 1992; Leake 1992b). I recommend all of them—they are the cream of the crop.

There are several people who deserve my thanks, whose input or support was critical in making this book a reality. I began collecting and collating much of the material in this book back in 1989 when Chris Riesbeck and I put together the first CBR Tutorial at National Conference on Artificial intelligence (AAAI). I thought it would be easy to compile a book from that; the material was essentially in place, and I had presented it many times. I was wrong. In any case, I thank Chris for all the material in here that came from our long discussions and from the transparencies we prepared. Several people have reviewed the manuscript and pieces of it along the way. Ray Bareiss and Eric Domeshek deserve the greatest thanks. Both were always available to read chapters fast and gave wonderful advice. Many other people read and commented on individual chapters: Bob Simpson, Tom Hinrichs, Mike Cox, Anthony Francis, Anna Zacherl, Ashok Goel, Ashwin Ram, Kris Hammond, Bill Mark, and Kevin Ashley come to mind right now. There were others. Nick Duncan and Joshua Bleier helped put the references together. There were also several anonymous reviewers along the way. Thanks to everyone. The deficiencies, of course, are all mine.

Much of the work reported in this book has been funded by individual National Science Foundation (NSF), Office of Naval Research (ONR), and DARPA contracts to the individuals working on the research. These agencies took the lead in making CBR a reality. The funding that paid for my projects and partially funded my time as I wrote the book has come from NSF (under grants IST-8608362 and IRI-8921256), ONR (under contract N00014-92-J-1234), Army Research Institute for the Behavioral Sciences (ARI) (under contract MDA903-90-K-0112), DARPA (under contracts F49620-88-C-0058, monitored by Air Force Office of Scientific Research (AFOSR), and N00014-91-J-4092, monitored by ONR), and IBM.

Now for thanking the most important people, my support network. First is the set of people who have helped me juggle all the things that I almost let fall while I was writing the book, especially in the past six months—my research scientists, Terry Chandler, Eric Domeshek, Mimi Recker, and Linda Wills, who took care of my technical life, and my assistants, first Jeannie Terrell and then Cindi Anderson, who kept everything else at work in order. Second, I want to thank Mike Morgan, who has been patient through all of the delays. He acted as if he

was sure I could do it when I wasn't so sure—I don't know if he meant it or not, but it helped. Unfortunately, it was only a few weeks ago, rather than a year ago, that I finally understood his best advice—"Make it perfect in the second edition." Third, I want to thank the members of the computing and cognitive science communities at Georgia Tech for not dumping more responsibilities on me in the past year than I could handle. Thanks to all of you.

Finally, I thank my husband, Mike, who has made things run around the house, and my kids, Joshua and Orly, who have tried their hardest to understand why I've had to be at work late, why they can't use the computer, and why I didn't make it to any soccer games. All have given their love and support throughout, especially at times when I needed it most. Thanks, guys—I love you and couldn't have done it without you.

Janet Kolodner
Atlanta, GA

FIGURE CREDITS

The publisher gratefully acknowledges permission to reproduce the following material:

Figure 2.1–2.4 From Hammond, K.J. (1989). *Case Based Planning: Viewing Planning as a Memory Task*, Academic Press, Boston, MA.

Figure 2.7 From Barletta, R. and Hennessy, D. (1989). "Case Adaptation in Autoclave Layout Design," in *Proceedings of the Case-based Reasoning Workshop*, DARPA, Hammond, K. (ed.).

Figure 2.9 From Dupuy, T.N. (1987). *Understanding War: History and Theory of Combat*, New York: Paragaon House.

Figure 2.10 From Ferguson, W. et al. (1992). "ASK Systems: An Approach to the Realization of Story-Based Teachers," *The Journal of the Learning Sciences*, Vol. 2, Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 95–134.

Figures 2.11 From Domeshek, E. and Kolodner, J. (1992). "A Case-based Design Aid for Architecture," in *Artificial Intelligence in Design '92*, Gero, J. (ed.), Kluwer Academic Publishers, Boston, pp. 497–516.

Figures 4.5, 4.6 From Kolodner, J.L. (1984). *Retrieval and Organization Strategies in Conceptual Memory: A Computer Model*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Figure 5.9 From Goel, A.K. et al. (1991). "Towards a Case-Based Tool for Aiding Conceptual Design Problem Solving," in *Proceedings of the Case-Based Reasoning Workshop*, DARPA, Bareiss, R. (ed.).

Figure 5.10 From Ferguson. W. et al. (1992). "ASK Systems: An Approach to the Realization of Story-Based Teachers," *The Journal of the Learning Sciences*, Vol. 2, Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 95–134.

Figure 5.11 From Domeshek, E. and Kolodner, J. (1992). "A Case-based Design Aid for Architecture," in *Artificial Intelligence in Design '92*, Gero, J. (ed.), Kluwer Academic Publishers, Boston, pp. 497–516.

Figures 5.12–5.15 From Simpson, R.C. (1985). *A Computer Model of Case-based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation*, Ph.D. Thesis, Technical Report No. GIT-ICS-85/18, School of Information and Computer Science, Georgia Institute of Technology.

Figure 5.16–5.18 From Koton, P. (1989). *Using Experience in Learning and Problem Solving*, Ph.D. Thesis, CS Department, Massachusetts Insitute of Technology.

Figures 5.21–5.24 From Goel, A. (1989). "Integration of Case-Based Reasoning and Model-Based Reasoning for Adaptive Design Problem Solving." Ph.D. Thesis, Department of Computer and Information Science, The Ohio State University.

Figure 5.25 From Dupuy, T.N. (1987). *Understanding War: History and Theory of Combat*, New York: Paragaon House.

Figure 5.26, 5.27 From Redmond, M.A. (1992). *Learning by Observing and Understanding Expert Problem Sovling*, Ph.D. Thesis, Technical Report No. GIT-CC-92/43, College of Computing, Georgia Institute of Technology.

Figures 6.2, 6.3, 6.4, 6.14 From Schank, R. and Osgood, R. (1990). *A Content Theory of Memory Indexing*, Technical Report No. 2, Institute for Learning Sciences, Northwestern University.

Figure 8.10 From Kolodner, J.L. (1984). *Retrieval and Organization Strategies in Conceptual Memory: A Computer Model*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Figures 9.8, 9.9 From Gentner, D. (1988). "Analogical Inference and Analogical Access." In *Analogica*, A. Prieditis (ed.). Los Altos, California: Morgan Kaufmann.

Figure 10.3 From Simoudis, E. (1991). *Retrieving Justifiably Relevant Cases from a Case Base Using Validation Models*, Ph.D. Thesis, CS Department, Brandeis University, Waltham, MA.

Figures 11.4–11.7 Adapted from Simpson, R.L. (1985). *A Computer Model of Case-based Reasoning in Problem Solving: An Investigation in the Domains of Disparate Mediation*, Ph.D. Thesis, Technical Report No. GIT-ICS-85/18, School of Information and Computer Science, Georgia Institute of Technology.

Figures 11.21, 11.22 From Koton, P. (1989). *Using Experience in Learning and Problem Solving*. PhD. Thesis, CS Department, Massachusetts Institute of Technology..

Figures 11.31–11.34 From Goel, A. (1989). "Integration of Case-Based Reasoning and Model-Based Reasoning for Adaptive Design Problem Solving." Ph.D. Thesis, Department of Computer and Information Science, The Ohio State University.

Figure 13.2 From Bareiss, E.R. (1989). *Exemplar-Based Knowledge Acquisition: A Unified Approach to Concept Representation, Classification and Learning*, Academic Press, Boston, MA.

Figures 13.3, 13.4 From Bareiss, E.R., Porter, B.W., and Weir, C.C. (1989). "Protos: An Exemplar-Based Learning Apprentice." *International Journal of Man-Machine Studies*, 29:549–561.

Figures 13.10–13.13 Courtesy of Cognitive Systems, Inc.

PART I

Background