



# KNOWLEDGE ENGINEERING AND MANAGEMENT

## The CommonKADS Methodology

A. Th. Schreiber, J. M. Akkermans,  
A. A. Anjewierden, R. de Hoog, N. R. Shadbolt,  
W. Van de Velde, and B. J. Wielinga

## **Knowledge Engineering and Management**

© 2000 Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

This book was set in Times Roman by the authors and was printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Schreiber, August Th.

Knowledge engineering and management: the CommonKADS methodology /

A. Th. Schreiber, . . . [et al.].

p. cm.

Includes bibliographical references and index.

ISBN 0-262-19300-0 (hc.: alk. paper)

1. Expert systems (Computer science). 2. Database management.

I. Schreiber, A. Th.

QA76.76.E95K5773 1999

006.3'32--dc21

99-19680

CIP

# Contents

---

<b>Preface</b>	<b>ix</b>
<b>1 Prologue: The Value of Knowledge</b>	<b>1</b>
1.1 The Information Society Is Knowledge-Driven . . . . .	1
1.2 Knowledge in Context . . . . .	3
1.3 Knowledge Engineering and Knowledge Systems . . . . .	5
1.4 Book Overview . . . . .	7
1.5 A Road Map for the Reader . . . . .	9
1.6 Bibliographical Notes and Further Reading . . . . .	11
<b>2 Knowledge-Engineering Basics</b>	<b>13</b>
2.1 Historical Perspective . . . . .	13
2.2 The Methodological Pyramid . . . . .	14
2.3 Principles . . . . .	15
2.4 Model Suite . . . . .	17
2.5 Process Roles . . . . .	20
2.6 Some Terminology . . . . .	22
2.7 Bibliographical Notes and Further Reading . . . . .	23
<b>3 The Task and Its Organizational Context</b>	<b>25</b>
3.1 Why Organizational Aspects Are So Important . . . . .	25
3.2 The Main Steps in Task and Organization Analysis . . . . .	27
3.3 The Feasibility Study: Organization Modelling . . . . .	28
3.4 Case: Social Security Services . . . . .	36
3.5 Impact and Improvement Analysis: Task and Agent Modelling . . . . .	44
3.6 Case: Ice-Cream Product Development . . . . .	50
3.7 Guidelines for the Context Modelling Process . . . . .	63
3.8 Bibliographical Notes and Further Reading . . . . .	66
<b>4 Knowledge Management</b>	<b>69</b>
4.1 Introduction . . . . .	69
4.2 Explicit and Tacit Knowledge . . . . .	69
4.3 The Knowledge Management Cycle . . . . .	71
4.4 Knowledge Management Has a Value and Process Focus . . . . .	72

4.5	Knowledge Management with CommonKADS . . . . .	75
4.6	Knowledge Management and Knowledge Engineering . . . . .	82
4.7	Bibliographical Notes and Further Reading . . . . .	83
<b>5</b>	<b>Knowledge Model Components</b>	<b>85</b>
5.1	The Nature of “Knowledge” . . . . .	85
5.2	Challenges in Representing Knowledge . . . . .	86
5.3	The Knowledge Model . . . . .	86
5.4	Domain Knowledge . . . . .	91
5.5	Inference Knowledge . . . . .	104
5.6	Task Knowledge . . . . .	112
5.7	Typographic Conventions . . . . .	117
5.8	Comparison with Other Analysis Approaches . . . . .	118
5.9	Bibliographical Notes and Further Reading . . . . .	121
<b>6</b>	<b>Template Knowledge Models</b>	<b>123</b>
6.1	Reusing Knowledge-Model Elements . . . . .	123
6.2	A Small Task Template Catalog . . . . .	128
6.3	Classification . . . . .	129
6.4	Assessment . . . . .	133
6.5	Diagnosis . . . . .	138
6.6	Monitoring . . . . .	143
6.7	Synthesis . . . . .	146
6.8	Configuration Design . . . . .	149
6.9	Assignment . . . . .	155
6.10	Planning . . . . .	159
6.11	Scheduling . . . . .	160
6.12	Task-Type Combinations . . . . .	165
6.13	Relation to Task and Organization Models . . . . .	166
6.14	Bibliographical Notes and Further Reading . . . . .	166
<b>7</b>	<b>Knowledge Model Construction</b>	<b>167</b>
7.1	Introduction . . . . .	167
7.2	Stages in Knowledge-Model Construction . . . . .	168
7.3	Knowledge Identification . . . . .	170
7.4	Knowledge Specification . . . . .	173
7.5	Knowledge Refinement . . . . .	181
7.6	Some Remarks about Knowledge-Model Maintenance . . . . .	184
7.7	Documenting the Knowledge Model . . . . .	184
7.8	Bibliographical Notes and Further Reading . . . . .	186

<b>8</b>	<b>Knowledge-Elicitation Techniques</b>	<b>187</b>
8.1	Introduction . . . . .	187
8.2	Characteristics of Knowledge Elicitation . . . . .	188
8.3	On Experts . . . . .	189
8.4	Elicitation Techniques . . . . .	191
8.5	An Elicitation Scenario . . . . .	201
8.6	Some Final Remarks . . . . .	213
8.7	Bibliographical Notes and Further Reading . . . . .	214
<b>9</b>	<b>Modelling Communication Aspects</b>	<b>215</b>
9.1	Role and Overview of the Communication Model . . . . .	215
9.2	The Communication Plan . . . . .	217
9.3	Case: Homebots — A Multiagent System for Energy Management . . . .	220
9.4	Transactions between Agents . . . . .	227
9.5	Detailing the Information Exchange . . . . .	229
9.6	The Homebots System Example Continued . . . . .	233
9.7	Validating and Balancing the Communication Model . . . . .	236
9.8	A Structured Process for Communication Modelling . . . . .	239
9.9	Bibliographical Notes and Further Reading . . . . .	240
<b>10</b>	<b>Case Study: The Housing Application</b>	<b>241</b>
10.1	Introduction . . . . .	241
10.2	Application Domain: Rental Residence Assignment . . . . .	241
10.3	Organization Model . . . . .	242
10.4	Task Model . . . . .	249
10.5	Agent Model . . . . .	253
10.6	Summary of Proposed Solution and Its Effects . . . . .	253
10.7	Knowledge Modelling . . . . .	253
10.8	Communication Model . . . . .	268
<b>11</b>	<b>Designing Knowledge Systems</b>	<b>271</b>
11.1	Introduction . . . . .	271
11.2	Structure-Preserving Design . . . . .	273
11.3	Step 1: Design System Architecture . . . . .	276
11.4	Step 2: Identify Target Implementation Platform . . . . .	279
11.5	Step 3: Specify Architectural Components . . . . .	282
11.6	Step 4: Specify Application within Architecture . . . . .	288
11.7	Design of Prototypes . . . . .	292
11.8	Distributed Architectures . . . . .	293
11.9	Bibliographical Notes and Further Reading . . . . .	294

<b>12 Knowledge-System Implementation</b>	<b>295</b>
12.1 Implementation in Prolog . . . . .	296
12.2 Implementation in Aion . . . . .	305
12.3 Bibliographical Notes and Further Reading . . . . .	314
<b>13 Advanced Knowledge Modelling</b>	<b>317</b>
13.1 Introduction . . . . .	317
13.2 Domain Knowledge . . . . .	318
13.3 Inference Knowledge . . . . .	331
13.4 Task Knowledge . . . . .	342
13.5 Bibliographical Notes and Further Reading . . . . .	345
<b>14 UML Notations Used in CommonKADS</b>	<b>347</b>
14.1 UML Background . . . . .	347
14.2 Activity Diagram . . . . .	348
14.3 State Diagram . . . . .	354
14.4 Class Diagram . . . . .	357
14.5 Use-Case Diagram . . . . .	368
14.6 General UML constructs . . . . .	370
14.7 A Small Case Study . . . . .	371
14.8 Bibliographical Notes and Further Reading . . . . .	376
<b>15 Project Management</b>	<b>377</b>
15.1 Control versus Flexibility: Striking the Balance . . . . .	377
15.2 Project Planning: The CommonKADS Life-Cycle Model . . . . .	381
15.3 Assessing Risks . . . . .	382
15.4 Plan: Setting Objectives through Model States . . . . .	383
15.5 Notes on Quality and Project Documentation . . . . .	386
15.6 Case: A Project on Nuclear Reactor Noise Analysis . . . . .	392
15.7 How <i>Not</i> to Manage a Knowledge-System Project . . . . .	399
15.8 Bibliographical Notes and Further Reading . . . . .	400
<b>Appendix: Knowledge-Model Language</b>	<b>403</b>
A.1 Language Conventions . . . . .	403
A.2 Language Syntax . . . . .	405
A.3 Full Knowledge Model for the Housing Application . . . . .	419
<b>Glossary of Graphical Notations</b>	<b>433</b>
<b>References</b>	<b>441</b>
<b>Index</b>	<b>447</b>

# Preface

---

## Aim of This Book

This textbook gives a basic but thorough insight into the related disciplines of *knowledge engineering* and *knowledge management*. Knowledge engineering is traditionally concerned with the development of information systems in which knowledge and reasoning play pivotal roles. Knowledge management is a recent area in business administration that deals with how to leverage knowledge as a key asset and resource in modern organizations. These two disciplines have strong ties. Managing knowledge within an organization is nowadays hardly conceivable without exploiting the vast potential of advanced information and knowledge systems. On the other hand, information system developers and knowledge engineers have come to realize that successful technical work is only possible if it is properly situated within the wider organizational context. Knowledge-engineering methods have thus gradually broadened their scope: they are not only used for knowledge-based systems development but have also shown their value in knowledge management, requirements engineering, enterprise modelling, and business process reengineering.

This book presents a comprehensive methodology that covers the complete route from corporate knowledge management to knowledge analysis and engineering, all the way to knowledge-intensive systems design and implementation, in an integrated fashion. This methodology, called “CommonKADS,” has been developed by a number of industry-university consortia over the past decade, and CommonKADS is nowadays in use worldwide by companies and educational institutions. The term “knowledge intensive” is intentionally vague, as it is often hard to define a strict borderline between knowledge-rich and knowledge-poor domains. In fact, most complex applications contain components that can be characterized as “knowledge intensive.” The applications need not at all be a “classic” knowledge-based system. Beyond information-systems applications, practice has shown that all projects in which knowledge plays an important role significantly benefit from the ideas, concepts, techniques, and experiences that come together in the CommonKADS methodology.

## Readership

This book is intended for practitioners and students in information systems engineering as well as in knowledge and information management. We assume that you are willing to consider new ways of managing the increasing complexity of information in applications

and organizations. In reading this book, it will be helpful if you have some background in information systems, have some understanding of information analysis or business process modelling, or have experience in the area of information management. The material of this book has proved to be useful for courses, tutorials, and workshops for industrial practitioners, as well as for advanced undergraduate and first-year graduate students in different information-systems related disciplines.

### **Unique Features of This Textbook**

With this book, we aimed to construct several bridges between traditionally different communities within the information-systems and knowledge-management areas:

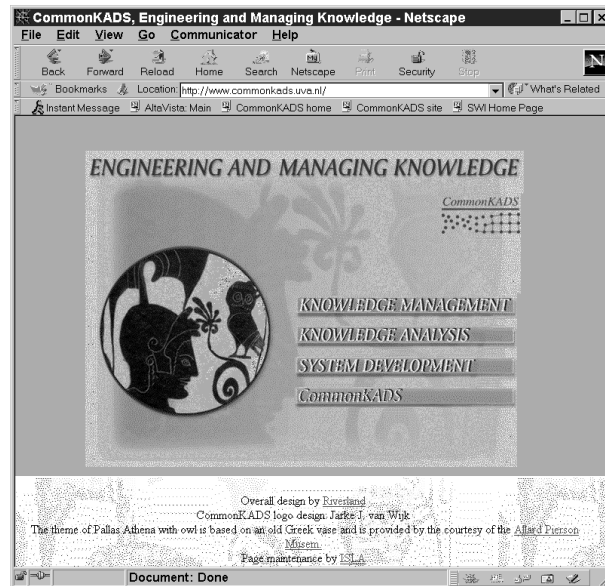
1. For information analysts and knowledge engineers, we show how knowledge analysis constitutes a valuable and challenging extension of established development approaches, particularly of object-oriented approaches such as the Unified Modelling Language (UML, Booch et al. 1998).
2. For knowledge managers, we show how a seamless transition and integration can be achieved from business analysis to information-technology (IT) systems modelling and design — a feature absent in almost all business process approaches, as well as systems-engineering methodologies.
3. For software engineers, we show how conceptual modelling of information and knowledge naturally provides the necessary baseline structures for reusable software architecture, systems design, and implementation.
4. For IT project managers, we show how one can solve the eternal dilemma of balancing management control vs. flexibility in a structured way that is directly based on quality systems development methodology.

Throughout the book, these points are illustrated by extensive case studies, which have been taken from real-life application projects carried out in different industries we have been working with over the years.

As a guide to readers with different specific interests, the first chapter contains a detailed road map to help you select those parts of the book that are most interesting and relevant to you.

### **Additional Material**

This book contains the consolidated baseline of the CommonKADS methodology. The material in this book is sufficient for readers to start useful work on knowledge-intensive applications. There is a wealth of additional material available, which could not be included in this book. For those who want to learn more about CommonKADS, this material



**Figure 1**  
Home page of the CommonKADS website at [www.commonkads.uva.nl](http://www.commonkads.uva.nl).

is accessible through the website at <http://www.commonkads.uva.nl> (see Figure 1). This website contains a large repository of additional CommonKADS information, including:

- exercises related to the material discussed in this book;
- case studies of applications;
- access to sample running systems;
- texts about additional modelling techniques, such as a dedicated formal specification language for knowledge systems;
- catalogs of knowledge-model elements developed in previous projects;
- pointers to support tools for CommonKADS, such as diagramming tools, elicitation-support tools, CASE tools, and parsers for the languages used.

## Background

CommonKADS is the product of a series of international research and application projects on knowledge engineering dating back as far as 1983. Historically, knowledge systems

developed mainly through trial and error. The methodological aspects received little attention, despite a clear need expressed by industry practitioners for guidelines and techniques to structure and control the development process. Accordingly, system developers and managers greatly appreciated the steps made by CommonKADS to fill this gap.

Over the years, the methodology has been gradually extended as a result of feedback from practitioners and scientists. Practical use of CommonKADS showed that many systems projects fail because of a technology-push approach. An organization can implement information and knowledge technology successfully only if both the system's role and its potential impact on the organization are made explicit, and are agreed upon before and during system development. Thus, the introduction of knowledge-oriented methods and techniques for organizational analysis represents a major advance. Organizational analysis aims at creating an application-pull situation. Such an approach provides assurance to users, clients, and stakeholders that a new system will actually solve a real problem or take advantage of a real opportunity within the organization. Other useful additions to the methodology deal with the modelling of complex user-system interaction; with the introduction of new specification techniques; and with the definition of a flexible, risk-driven, and configurable life-cycle management approach that replaces the waterfall model for information-systems projects, as classic as it is overly rigid.

## Experiences

Early on, companies began using the knowledge technology products provided by CommonKADS. This contributed greatly to the products' success. As early as 1986, the Dutch company Bolesian Systems, now part of the large European software firm Cap Gemini, exploited the first version of CommonKADS and refined it into their in-house method for knowledge-systems development. They have built a very large number of commercial systems, mostly in the financial sector. More recently, the Everest company is making use of CommonKADS in a similar manner. Many banks and insurance companies in the Netherlands have systems developed with CommonKADS in daily use for assessing loan and mortgage applications. In Japan, several big companies, including IBM, are using CommonKADS in their in-house development; for example, to increase software-architecture reusability. A well-known application in the UK is the credit card-fraud detection program developed by Touche Ross Management Consultants for Barclay Card. All the "Big Six" worldwide accounting and consultancy firms have integrated smaller or larger parts of CommonKADS into their proprietary in-house development methods.

CommonKADS also frequently serves as a baseline for system development and research projects, such as the European IT programme and national government projects. Furthermore, the CommonKADS methods are nowadays in use for purposes other than system development, such as knowledge management, requirements capture, and business-process analysis. The US-based Carnegie Group, for example, has applied CommonKADS

in this way in a project for US West. Likewise, the Unilever company uses CommonKADS as its standard both for knowledge-intensive systems development and for knowledge management.

## The Authors

Because it is difficult to write a textbook with many different authors, we decided early on that it would be best if only two authors actually wrote the text, and that the others contributed to the drafts. Of course, the material contains ideas and is based on the work of *all* the authors. Accordingly, Guus Schreiber has been responsible for the general editing process and for chapters 5-7 and 10-14, whereas Hans Akkermans wrote most of chapters 1-4, 9 and 15. Nigel Shadbolt contributed chapter 8, Robert de Hoog wrote part of chapter 4, and Anjo Anjewierden checked the CommonKADS Conceptual Modelling Language examples and contributed the appendix.

## Acknowledgments

Many persons, companies and institutions have contributed to the CommonKADS knowledge methodology in its various stages of development.

The TRACKS workshop participants, who acted as an international industrial review board for the book, are gratefully acknowledged for their extensive reviews of the material. Their suggestions have led to many improvements in the book. We are particularly grateful to Unilever Research Laboratories in Vlaardingen (The Netherlands), also for their financial support. This book has profited from the input and extensive feedback from many colleagues at Unilever. The CUPIDO platform, a group of companies and institutions in the Netherlands and Belgium, provided many forms of support for this book. Members of CUPIDO are Bolesian, CIBIT, Everest, Platinum, Unilever, and the Royal Dutch Navy. The Aion implementation described in Chapter 12 was developed by Leo Hermans and Rob Proper of Everest. The diagnosis model in Chapter 6 is based on input from Richard Benjamins. The scheduling model in Chapter 6 was kindly provided by Masahiro Hori of IBM-Japan. Brice LePape of the ESPRIT office of the European Commission has supported the development of CommonKADS from the beginning. In the final stages we received funds from the Commission through the TRACKS project (ESPRIT P25087) in order to organize a number of workshops to get feedback on the draft book. This project also supported the construction of the website with additional material. Rob Martil, then at Lloyd's Register, now at ISL, contributed to the development of the CommonKADS model suite and to the ideas about project management. Many ideas about knowledge management (Chapter 4) were developed in cooperation with Rob van der Spek of CIBIT, while teaching CommonKADS to many students with a conventional systems development background at CIBIT helped in clearly focusing ideas about the worldview behind CommonKADS.

Joost Breuker was one of the founding fathers of KADS and developed, together with Paul de Greef and Ton de Jong, the typology for transfer functions (Chapter 5). Peter Terpstra contributed to foundations of the work on knowledge system design and implementation (Chapter 11). The work of Klas Orsvarn (SICS) and Steve Wells (Lloyd's Register) is acknowledged as an inspiration for the description of the knowledge-modelling process (Chapter 7). Kieron O'Hara provided valuable comments and input for Chapter 8. The work of Frank van Harmelen and Dieter Fensel on formal languages for knowledge-model specification were an important source of information. We are grateful for discussions with Ameen Abu-Hanna, Gertjan van Heijst, Wilfried Post, and Annette ten Teije during their Ph.D. and post-doctoral work at SWI. Wouter Jansweijer and other people working on the KACTUS project contributed to the ideas underlying the material on advanced knowledge modelling (Chapter 13). Marc Linster The first author, Guus Schreiber, is grateful for the feedback given by his course students in "Data and Knowledge Modelling," who had to study from draft material, and whose experiences have proved valuable in increasing the quality of the book. The second author, Hans Akkermans, would like to thank Rune Gustavsson (University of Karlskrona-Ronneby, Sweden) for his support and feedback on courses and workshops given in Sweden based on the draft textbook, and to Fredrik Ygge, Alex Ratcliffe, Robert Scheltens, Pim Borst, Tim Smithers, Amaia Bernaras, Hans Ottosson, Ioa Gavrilă, Alex Kalos, Jan Top, Chris Touw, Carolien Metselaar, Jos Schreinemakers, and Sjaak Kaandorp for providing case study material and/or giving their feedback on Chapter 1, Chapter 3, Chapter 4, Chapter 9 and Chapter 15. Jacobijn Sandberg provided feedback based on the use she made of CommonKADS in other projects. Machiel Jansen helped to develop the classification model in Chapter 6. Xavier Nicaise and Axel Wegner of ARTTIC played an important role in organizing the support actions for this book. Saskia van Loo and Lenie Zandvliet provided many forms of organizational support. Most figures in this book were drawn with Jan Wielemaker's MODELDRAW program. The CommonKADS logo (elements of which are used in the book cover) was designed by Jarke J. van Wijk.

The research on methodology described in this book has been supported by a number of projects partially funded by the ESPRIT Programme of the Commission of the European Communities, notably projects P12, P314, P1098 (KADS), P5248 (KADS-II), P8145 (KACTUS), and P25087 (TRACKS). The partners in the KADS project were STC Technology Ltd. (UK), SD-Scicon plc. (UK), Polytechnic of the South Bank, (UK), Touche Ross MC (UK), SCS GmbH (Germany), NTE NeuTech (Germany), Cap Gemini Innovation (France), and the University of Amsterdam (the Netherlands). The partners in the KADS-II project were Cap Gemini Innovation (France), Cap Gemini Logic (Sweden), Netherlands Energy Research Foundation ECN (the Netherlands), ENTEL SA (Spain), IBM France (France), Lloyd's Register (UK), the Swedish Institute of Computer Science (Sweden), Siemens AG (Germany), Touche Ross MC (UK), the University of Amsterdam (the Netherlands), and the Free University of Brussels (Belgium). The partners in the KACTUS project were Cap Gemini Innovation (France), LABEIN (Spain), Lloyd's Register (United Kingdom), STATOIL (Norway), Cap Programmatore (Sweden), University of Amsterdam (the Netherlands), University of Karlsruhe (Germany), IBERDROLA (Spain), DELOS (Italy), FINCANTIERI (Italy), and SINTEF (Norway). The partners in the TRACKS project were Intelligent Systems Lab Amsterdam (the Netherlands), AKMC Knowledge Management BV (the Netherlands), Riverland Next Generation (Belgium), and ARTTIC (Germany). This book reflects the opinions of the authors and not necessarily those of the consortia mentioned.