INTELLIGENT SYSTEMS DESIGN

Integrating Expert Systems, Hypermedia, and Database Technologies

INTELLIGENT SYSTEMS DESIGN

Integrating Expert Systems, Hypermedia, and Database Technologies

Larry Bielawski
and
Robert Lewand
Goucher College
Baltimore, Maryland

John Wiley & Sons
New York Chichester Brisbane Toronto Singapore

DS4/6

In recognition of the importance of preserving what has been written, it is a policy of John Wiley & Sons, Inc., to have books of enduring value published in the United States printed on acid-free paper, and we exert our best efforts to that end.

Library of Congress Cataloging-in-Publication Data:

Bielawski, Larry.
Intelligent systems design : expert systems, hypermedia, and database technologies / Larry Bielawski and Robert Lewand.
p. cm.
Includes bibliographical references.
1. Expert systems (Computer science) 2. Hypermedia systems.
3. Data base management. I. Lewand, Robert. II. Title.
QA76.76.E95855 1991
006.3—dc20 90-44334
CIP

ISBN 0-471-52958-3 (cloth). — ISBN 0-471-52535-9 (pbk.)

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Preface

Intelligent systems can take many forms. Natural-language translation systems, computer chess games, robotics applications, vision- and speech-recognition systems, neural networks, and knowledge-based or expert systems are all special types of intelligent systems. What each of these systems offers to a varying degree is the ability to represent artificially the human mind at work. In attempting to replicate human intelligence, these programs often strive to manipulate vast amounts of information, to offer some degree of problem-solving ability, and to augment a human's ability to connect both information and concepts in nonlinear ways.

In this book, we focus primarily on intelligent systems that are based on the integration of expert systems, hypermedia, and data-base technologies, for together they offer a *rich* environment for creating computer applications that can increase productivity enormously and act as intelligent assistants. Furthermore, though hypermedia and expert systems technologies date back thirty or more years as independent technologies, it has only been in the last two or three years that their paths have fully converged, offering system developers a flexible environment that takes advantage of existing information and data. In Appendix A we describe a brief history of how expert systems and hypermedia evolved independently and then finally merged in the mid-1980s.

The melding of hypermedia and expert systems into the broader class of intelligent systems applications has also brought attention to the fact that often no one technology will completely solve a given problem, so the need to integrate diverse technologies within a single application is both common and necessary. When looked at from a DP/MIS perspective, this combination of technologies is quite attractive as it leads to overall increases in productivity. Indeed, when hypermedia and expert systems technologies are merged with information systems and data-base technologies, a synergistic effect is realized, offering more useful applications than could have been achieved with either technology alone. And it is this integral relationship that we focus on in this book as we explore and document the integration of hypermedia, expert systems, and data-base technologies in creating intelligent applications.

Besides including detailed definitions of expert systems and hypermedia, we also provide an overall methodology for integrating these technologies with existing information and demonstrate this process with three different models. The experiments and case studies that are used for illustration in this book are derived from our work and that of others in the field of intelligent systems design. In each case, we provide explicit details of how each application was inspired and how hypermedia and expert systems technologies fused with existing information and data into a single coherent application. Our hope is that other application developers, whether they are area managers, computing professionals, or domain experts, will gain a better overall understanding of these emerging technologies. At the same time, we hope that our readers acquire the necessary skills and knowledge that will allow them to successfully integrate hypermedia and expert systems technologies into useful intelligent systems applications.

In this process, an important intermediate goal is to acquaint readers with select tools on the market that allow system developers to easily integrate the technologies discussed without the need to program in conventional computer languages such as C, Pascal, LISP, or PROLOG. We have found that a great deal of development work in the area of intelligent systems can be accomplished on desktop systems with low-cost tools. Reasons for this include the increasing power of Intel 80386-based PCs and Apple Macintosh II workstations and the abundance of easy-to-use development tools that cut down development time, usually by several orders of magnitude. Another reason we have deliberately focused on PC and Macintosh-based development tools or "shells" concerns how intelligent systems are being deployed on a smaller and smaller scale. As Adam Osborne, former president of Paperback Software put it, AI's "real place is in a myriad of simple, little applications" (Osborne 1988). Intelligent systems offer the possibility of being helpful not for what they are (AI-based applications), but for how these programs assist professionals in doing what they do best, typically augmenting productivity with minimal up-front investments.

We believe that current strategic corporate moves toward using hypermedia and expert system technologies as a leveraging technology within traditional information systems and data-base applications indicates continuing widespread adoption of intelligent systems. According to Alain Rappaport of Neuron Data Inc., "By providing a common data structure, the AI tool acts as a technological glue between applications. AI becomes to software what networking is to hardware—a subtle, very powerful enabling technology" (Rappaport 1989). Because combining hypermedia and expert systems technologies can supply embedded AI-based functionality within existing applications to leverage data or information, intelligent systems often add value to a program or application, rendering it more competitive. In fact, according to some industry experts, "Embedded AI will become so useful that, by 1998 or 1999, a vendor will be unable to successfully sell a product without a far greater form of embedded artificial intelligence than we have today" (Leininger 1988). For AI-based intelligent systems to achieve this goal, they must take into account existing real-world needs for information processing. Moreover, they must encompass a broader range of functions than expert-systems technology currently does alone. Two such functions are nonlinear navigation through extensively linked information (in both graphical and text form) and intelligent location and retrieval of information from common sources, such as data-base systems.

The combination of these two functions creates the possibility for discussing the integration of hypermedia, expert systems, and data-base technologies within the context of a single book. Books that deal with these distinct technologies exist in abundance, but none addresses their integral relationship and provides models for their seamless integration. *Intelligent Systems Design,* therefore, is unique in its overall breadth and scope: readers will first get clear definitions of expert system and hypermedia technologies and then acquire a workable design/development methodology, see them in action within six case studies, gain valuable integration/engineering skills, and review commonly available development tools or shells. We hope that this approach not only will result in a rewarding reading experience but will foster the development of useful intelligent systems in organizations worldwide.

LARRY BIELAWSKI
ROBERT LEWAND

Baltimore, MD
January 1991

Acknowledgements

We are greatly indebted to the tool vendors and application developers who have allowed us to discuss their work or products within this book. Special thanks go to the people at AI Corp, Knowledge Garden, Cognition Technology, IntelligenceWare, Human Intellect Systems, Information Builders, Peridom, Paperback Software International, and Neuron Data for allowing us to review their intelligent system building tools. We also want to thank members of the Food and Agriculture Organization of the United Nations, the National Agricultural Library, Zenith Data Systems, Westinghouse, Ford Motor Company, The Carnegie Group, Hughes Aircraft, and RWD Technologies for their help and permission to use their fielded systems or prototypes for examples. We also thank the many tool vendors who made copies of their products available to us for review and offered their personal assistance and documentation in helping us to better understand their products' contribution to the field of intelligent systems. We also wish to thank Diane Cerra and Terri Hudson at John Wiley & Sons, Inc., for sharing our vision and enthusiasm for *Intelligent Systems Design*, and finally Jaye Crooks and Lynda Hamilton for their help in preparing the final manuscript.

About the Authors

Larry Bielawski and Robert Lewand are educators, authors, and applications developers in the emerging field of intelligent systems. Their work has been chronicled in the case studies in Chapter 7 of this book and in trade journals such as *Computerworld*, *InfoWorld*, and *PC Week*. Between them they have over twenty years of experience as computer professionals and have served as trainers and consultants to numerous businesses and organizations. Their most recent work with the United Nations, the National Agricultural Library, Zenith Data Systems, Texas Instruments, John Wiley & Sons, and others has given them the opportunity to fully explore the emerging technologies discussed in the text and to be among the first to gain direct experience with integrating them into innovative applications. Both of Goucher College in Baltimore, Maryland, Larry Bielawski is Director of the Decker Center for Information Technology, and Robert Lewand is Professor of Mathematics and Computer Science.

Contents

SECTION I

Intelligent Systems Concepts

In this first section, we build the foundation for our discussion of intelligent systems. Chapter 1 defines the concept of an intelligent system, describing what such a system contains and how it behaves. Chapter 2 lists several types of problems for which intelligent systems are particularly well-suited and then goes on to describe the architecture of both of these technologies and to introduce the notion of an intelligent system development tool. If you are already familiar with expert systems and hypermedia concepts, you may wish to regard Chapter 2 only as a reference chapter; if you are not well-versed in both technologies, you will find the chapter quite helpful in understanding the remainder of the book. By reading Section I, you will understand exactly what intelligent systems are, the purpose they serve, and their architecture.

1

An Overview of Intelligent Systems

In the Preface, we began our discussion of intelligent systems by stating that unlike traditional computer programs intelligent computer applications attempt to emulate the human thinking process and can serve therefore as an extension of our creative and problem-solving abilities. More specifically, intelligent systems represent a combination of technologies that attempt to parallel or replicate human behavior within specific and narrowly defined contexts. These technologies have their basis, mostly, in artificial intelligence research and development. To facilitate a better understanding of the nature of intelligent systems, we begin with the general goals and objectives of artificial intelligence (AI).

ARTIFICIAL INTELLIGENCE AND INTELLIGENT SYSTEMS

According to R. J. Brachman, a pioneer in the AI community:

> A widely recognized goal of artificial intelligence is the creation of artifacts (usually software programs) that can emulate humans in their ability to reason symbolically, as exemplified in typical AI domains such as planning, natural language understanding, diagnosis, and tutoring. Currently most of this work is predicated on the belief that intelligent systems can be constructed from explicit, declarative knowledge bases, which in turn are

> operated on by general, formal reasoning mechanisms. This fundamental hypothesis of AI means that knowledge representation and reasoning—the study of formal ways of extracting information from symbolically represented knowledge—is of central importance to the field. (Brachman 1988)

Brachman's description of AI relates to our general definition of intelligent systems in two ways. First, it stresses knowledge representation and reasoning at the heart of any system that reflects intelligence, and, second, it suggests that the only way such systems can behave in this manner is if they contain *formal* mechanisms for representing knowledge and employing inference techniques that model conventional, computationally based systems.

For example, if an intelligent system is going to troubleshoot a telephone network switching unit, an automobile's fuel injection system, or a faulty hydraulic pump, it must first work from a body of experience and information indigenous to the domain (knowledge representation). Then it can employ a diagnostic method (based on inference) to proceed from known symptoms (facts) to probable causes and corrective measures (expert advice). Central to this type of intelligent activity is the ability not only to work logically but to access and synthesize discrete pieces of information in creating a new understanding of the problem situation and its possible resolution. This definition and elementary example of artificial intelligence technology points to the need to distinguish intelligent systems from conventional computer programs.

INTELLIGENT SYSTEMS AND CONVENTIONAL PROGRAMS

The behavior and attributes of an intelligent system distinguish it from conventional computer programs. Traditional computer applications, such as databases, spreadsheets, graphics programs, and text-processing programs, are fundamentally time-saving tools that have replaced manual approaches to tasks such as calculating, sorting, typing, drawing, and so on. They do not attempt to replicate human activities such as problem solving, diagnosis, and planning. Furthermore, such programs simply employ algorithms and exhibit no intelligence *per se.* These algorithms typically create one solution path, require a complete set of data, and use a predictable set of steps defined by the programmer. It could be argued that a computer is acting intelligently when it performs a computation or a comparison. This definition of intelligent behavior, however, is too broad to be of much value since here the computer is not using knowledge or inference to reach a result, but rather employing a prescribed sequence of steps that do not vary even if environmental conditions change.

Intelligent systems, in comparison, are more flexible and adaptive in that they draw on knowledge and the power of association and inference to steer the direction of a running program toward useful results. They do so by dealing effectively with "the complex interaction of many factors that must be considered as a whole, rather than as a series of steps" (Scown 1985), a process often referred to as "symbolic processing." Scown points out that AI-based programs can not only manipulate these symbols but govern the relationships among the symbols, which can, in turn, represent real-world entities; that is, unlike conventional programs that manipulate variables whose values are known, intelligent systems can manipulate symbols independent of their values. They do so by having rules within the system that govern or manipulate the relationships among the symbols. Consequently, even when working with limited information, intelligent systems can still reach a useful result. In fact, Scown suggests that a system that is only 30–50 percent complete can still be quite useful as a prototype, whereas this is not the case with any conventional program.

Because intelligent systems can manipulate symbols that represent real-world entities, they are capable of working with knowledge as well. To understand this principle, we must distinguish among data, information, and knowledge. As Scown points out:

> We can think of data as any value available to the system for processing. Information can be described as data that has been selected and organized for a particular purpose. Knowledge, in the realm of artificial intelligence, is information structured in a way that brings out and exploits the relationships among the pieces of data. The distinctive aspect of the AI approach is the emphasis on the storage and manipulation of the relationships among symbols. (Scown 1985)

The key factors in intelligent systems are:

- The ability to use knowledge to perform certain tasks or solve problems, and
- The capacity to exploit the powers of association and inference in attempting to deal with complex problems that resemble the real world.

These primary characteristics are helpful in defining intelligent systems because they not only suggest how they might behave but point to the mechanisms and techniques they usually contain.

THE ROLE OF INTELLIGENT SYSTEMS

Among our intelligent activities as humans is the ability to store and retrieve vast amounts of information efficiently, to solve complex problems or reach decisions, and to connect our thoughts and ideas in nonlinear, associative ways. Central to this kind of behavior is not only our complex organizational skills but our ability to adapt or modify our behavior based on reason and to employ several unique skills given the situation at hand.

For computer systems to be "intelligent," they must possess at least a subset of these abilities as they model specific human-related tasks. Moreover, intelligent systems need not act as independent agents, replacing human experts in a given situation. Rather, they can function as intelligent assistants, augmenting or supplementing human expertise while increasing productivity. Some characteristics of intelligent systems are:

- Intelligent systems behave logically.
- Intelligent systems solve complex problems.
- Intelligent systems are responsive and adaptive.
- Intelligent systems provide nonlinear program navigation.
- Intelligent systems make effective use of existing information.
- Intelligent systems are user-friendly and highly interactive.

Looked at collectively, the listed characteristics are a tall order for any intelligent system. Indeed, if such systems were based on commonsense knowledge and used in broad application areas, they would surely provide limited results. However, when applied in a narrow field or domain, intelligent systems can be very effective and can exhibit many of the characteristics, which will be described in more detail below.

Intelligent Systems Behave Logically

By using either inductive or deductive methods, intelligent systems must be able to represent knowledge and to reason effectively based on logical inference, such as the recognition of causal relationships. For example, intelligent systems are typically capable of deducing that X is a subset of Z given that X is a subset of Y and Y is a subset of Z.

In other instances, an intelligent system may reach some generalizations based on factual information related in a set of examples. Drawing on the principles of induction, such systems possess the ability to take a "collective look" at the information given in the form of examples, transforming it into knowledge. For instance, a marketing analyst might be able to forecast future sales directions based on current patterns or trends. An intelligent system should, in a limited domain, be able to parallel this type of behavior to some degree.

Intelligent systems will, therefore, contain mechanisms for formally representing knowledge and employing a reasoning method. The most common forms of knowledge representation are production rules, decision trees, and semantic networks, which are defined in detail in Chapters 2 and 3. The reasoning processes reflect inductive or deductive methods and incorporate