Article

Internet-based expert systems

Ralph Grove

James Madison University, Department of Computer Science, Harrisonburg, VA 22807, USA
E-mail: rfgrove@computer.org

Abstract: *The Internet offers a large potential for delivery of various information-based services, including the services of intelligent applications. As the availability of the Internet has grown, its value as a medium for the delivery of expert systems in particular has increased. There are now a large number of expert systems available on the Internet, including applications in industry, medicine, science and government. Though the Internet provides several advantages for expert system development, it also presents some special problems. These advantages and disadvantages are explored in more detail in this paper. The paper also presents a review of several Internet-based expert systems with a representative sample of publicly available applications, and a discussion of typical tools for developing Internet-based expert systems. A case study of an Internet-based expert system is presented as well.*

Keywords: expert systems, World Wide Web, Internet, Web-based information systems

1. Introduction

Expert system technology was initially developed in the mid-1960s, though the roots of research and development in this area go back to the notion of production systems from two decades earlier. Classical applications from the first generation of expert systems include DENDRAL, an expert system for determining the structure of organic molecules, and MYCIN, an expert system for assisting physicians with diagnosis of blood disorders. The first wave of commercial expert systems applications came a decade later, beginning with Prospector, a geology expert system for locating mineral deposits, and R1/XCON, an expert system developed by Hewlett Packard for configuring hardware orders. The design of these applications is primarily rule-based, with additional facilities for handling uncertainty, for explanation of inferences, and for knowledge acquisition. Architecturally, these applications were stand-alone systems based on mainframe, minicomputer or personal computer platforms, or LAN-based distributed applications.

Despite their commercial success, several problems are often associated with conventional expert systems, including the following. One problem is *availability*, i.e. putting the expertise at the place and time where it is needed. An expert system having an interface that is located in a physician's office, for example, will be of little use to the physician working under pressure at a patient's bedside. For many users, even a slight inconvenience in access is enough of an obstacle to prevent acceptance of an expert system. Another problem is *software distribution*. As knowledge base and user interface components of stand-alone applications are updated, those updates must be physically distributed to all users, along with necessary documentation and instructions. This can require many separate software installations and upgrades over time, often beyond the competence of users. The resulting record-keeping problem for project managers is of significant difficulty as well. A third problem is *communication*. A lack of common protocols for the exchange of knowledge among expert systems has discouraged designs involving cooperation or dynamic information sharing among applications. The resulting isolation is a factor in the brittleness of expert system products, which tend to have a tightly circumscribed area of competence.

2. Expert systems on the Internet

2. 1. Rationale for expert system delivery via the Internet

There are several factors that combine to make the Internet, by contrast to traditional platforms, a more effective base for expert system delivery.

The Internet is widely accessible Internet usage continues to grow rapidly, with varying estimates putting the number of Internet users at around 100 million in the USA and 200 million worldwide (NUA Internet Surveys, 2000). Business Intranets (private Internet provider (IP) networks) provide access in many workplaces and business Extranets (private business-to-business IP networks) are becoming more common in support of business transactions. Wireless communication makes Internet access completely portable as well. Due to the ubiquity of the Internet, Internet-based expert systems can be accessed literally anywhere in the world, at the precise location at which they are needed. If NASA's plans for an extraterrestrial network are successful, that availability will extend to outer space as well (Charles, 1999). A Mars-based cargo ship, for example, could initiate a diagnostic session with an expert system located at the factory on Earth at which it was designed.

Web-browsers provide a common multimedia interface HTML-compatible browsers are installed on virtually all desktop workstations and personal computers. The common controls and format dictated by HTML provide a standard user interface platform on which developers can build. The presence of a standard user interface framework not only simplifies development efforts but also greatly reduces user training and support requirements for expert system developers. Multimedia (graphics, text, sound etc.) can be easily incorporated into user interfaces based on Web browsers when appropriate.

Several Internet-compatible tools for expert system development are available There are a variety of expert system development tools with features that facilitate Internet-based development, including server components, HTML interfaces, and compatibility with Internet protocols and languages. These include commercial development environments and free expert system shells. A sample of these tools is described in more detail below.

Internet-based applications are inherently portable The Internet provides a development environment that is platform independent and widely available. For Internet-based expert systems, this means that there is no need for special distribution and installation of expert system software in advance of its use. Rather, applications are provided on demand at the time and place they are needed. Furthermore, applications can easily be relocated (to different servers), upgraded and modified without affecting users directly.

Emerging protocols support cooperation among expert systems Protocols such as CORBA, DCOM and Jini provide standard mechanisms for the exchange of information and services among applications. Though these protocols support data and service sharing through object interactions, they must still be adapted to the expert system application level in order to enable cooperation among intelligent processes. XML is another protocol that offers potential for interaction among expert systems, by providing a means to develop common knowledge interchange languages.

2. 2. Survey of expert systems on the Internet

The following summary is drawn from descriptions of Internet-based expert systems in recent technical literature as well as descriptions found on Web sites associated with expert system applications. These applications fall into four distinct application domains, including business/industry, education/research, government and medical informatics.

Business and industry

- The Configurate and Sizing Tool/SAP (CAST) was developed by Hewlett Packard to aid in configuring SAP business information warehouse implementations (Hewlett Packard, 2000). CAST is a rule-based expert system built with Multilogic's ExSys expert system tool.
- The Coating Alternatives Guide (Research Triangle Institute, 2000a) and Solvent Alternatives Guide (Research Triangle Institute, 2000b), both developed by the Pollution Prevention Program at the Research Triangle Institute, have the goal of reducing industrial pollution. Both systems are rule-based expert systems, developed in a custom shell written in ColdFusion, a Web application server framework.

Education and research

- The Douglas Fir Cone and Seed Insects System (Acquired Intelligence, 2000) identifies pests that attack Douglas Firs. The system incorporates knowledge in the form of rules and action tables. It was built using Acquire, a commercial expert system shell.
- The Expert System for Thermodynamics (TEST), developed at San Diego State University (Bhattacharjee, 2000), assists in solving problems in thermodynamics. TEST is designed as an HTML-based decision tree that incorporates Java applets to perform calculations and what-if scenarios.
- The Reptile Identification Helper (Grove & Hulse, 1999) is a Web-based expert system for identifying amphibian and reptile specimens found in the state of Pennsylvania, as part of an ongoing biological census. The Reptile Identification Helper is designed as a client–server application, incorporating Java Expert System Shell as the server-side inference engine and a multimedia interface in HTML and Javascript.

Government

- The OSHA Hazard Awareness Advisor (HAA) is an expert system that recognizes industrial hazards (Stern, 1999). The HAA interrogates the user about the user's workplace situation and generates a report concerning possible hazards and related OSHA rules. The HAA was developed using Multilogic Exsys.

- An expert system for optimizing the selection of landfill siting has been developed recently in Taiwan (Kao, 1998). The system employs a complex rule base incorporating environmental, sociocultural, engineering and economic rules derived from publications and case studies. The design also includes a fuzzy model of groundwater behavior.

Medicine

- Willard *et al*. (1995) describe an Internet-based medical information system for a clinical setting that includes an expert system for ordering laboratory tests for patients. The expert system component is a custom design developed using an object-oriented variant of the scripting language Tcl.
- HEPAXPERT/WWW is an Internet-based interface to an expert system for interpreting serologic tests for hepatitis infections (Chizzali-Bonfadin *et al*., 1995). In this case, test results are entered via the Web, while processing is done off-line and conclusions are e-mailed to the user.
- The Protocol Assistant is an Internet-based expert system for diagnosis of parotid tumors (Simpson *et al*., 1998). Knowledge is expressed in PROForma (a language for representing best practice guidelines) and is then translated to rules in Java Expert System Shell. The entire system is packaged as a Java applet for downloading and client-side execution.
- Riva *et al*. (1997) describe an Internet-based system for intelligent management of the treatment of diabetic patients. This application interacts with both patients and doctors as it collects data from patients and makes recommendations to doctors about therapy. The design includes fuzzy logic control, a LISP Web server and client-side Java and JavaScript components for performing calculations.
- A decision support system to support the ear, nose and throat unit of a primary health care center is described by af Klercker and af Klercker (1998). The system is designed as a decision tree, derived by induction from actual past cases, and is implemented in PERL modules that dynamically generate HTML.

2. 3. *Examples of publicly accessible expert systems on the Internet*

The following Internet-based expert systems are publicly accessible. These three applications, from business, government and research, respectively, are typical in design and operation of the applications mentioned above.

- The Coatings Alternative Guide (CAGE) is an expert system designed to help users of industrial coatings select alternatives that are less hazardous and more environmentally friendly. URL: http://cage.rti.org
- Several expert systems dealing with workplace safety are available at the OSHA Advisors and Computer-based Training page. URL: http://www.osha-slc.gov/dts/osta/oshasoft
- The Reptile Identification Helper is part of the Pennsylvania Herpetological Atlas, an ongoing census of reptile and amphibian species in Pennsylvania. This expert system assists users in the identification of specimens encountered in the field. URL: http://www.nsm.iup.edu/pha/rih

3. Design issues

3. 1. *Architecture, scalability and related problems*

Several design patterns have emerged from development and research efforts involving the Internet and intelligent applications. *Web-based expert systems* are based on traditional expert system technology, rule-based and case-based reasoning primarily; they have been adapted from organic designs to Internet use by incorporating client–server architectures and Web browser based interfaces. The inference engine usually executes on the server side, though small applications may download as Java applets and run on the client side. Simple decision trees may be implemented strictly in HTML as well, using hyperlinks. *Intranet-based expert systems* are similar except that access is restricted to a private network for security, privacy or legal reasons. Medical applications, for example, are subject to stringent privacy requirements that may require a closed network. *Intelligent agents*, alternatively referred to as softbots or knowbots, are highly intelligent applications active on the Internet. Though the term lacks an accepted and precise definition, it is generally reserved for applications that have some degree of reactivity, autonomy and adaptability (Sycara, 1998). In this sense, expert systems are typically less complex than intelligent agents, but an expert system may be a component of a larger agent architecture. *Cooperating expert systems* are applications that seek to transcend the brittleness normally associated with narrowly focused expert systems and to synthesize larger inference abilities by allowing expert systems to share their expertise. Work in this area is still in the exploratory stages, dealing with problems such as knowledge formats, reliability, cost recovery and communication protocols.

The scalability of Internet-based expert systems is significant in several different dimensions. One dimension is the rule base size and complexity. Theoretically, the process of matching rules and knowledge grows exponentially with size, as each new fact and rule antecedent requires matching to all existing components. In fact, however, modular knowledge programming techniques and efficient inference algorithms such as the Rete algorithm have mitigated this problem. The continual exponential increase in host processing power (Moore's law) has allowed corresponding growth in applications as well. Of more significance is the growth in a second dimension, the user base of expert system applications on the Internet, both in terms of numbers and in the degree of geographic distribution. For publicly available expert systems with general application, there is a potential for large growth in the number of users as more of the world's population gains access to the Internet. The primary solution to this problem of growth will be more processing power to support additional inference engines, which are easily replicable assuming that the users are independent, which is normally the case. The geographic dispersal of users is potentially more problematic, since the traversal of distance through TCP/IP is not simply a function of electronic speed, but requires additional processing time along the route (in the hypothetical interplanetary realm of networking, electronic delay becomes an issue as well). Increased bandwidth and speed in global communication channels is one solution, though there is no promise of increases sufficient to eliminate the problem. The same issue is a problem in the third dimension of growth, which is the degree of cooperation between heterogeneous expert systems. Practical applications of cooperating expert systems must deal with the issue of communication delay, along with other issues mentioned earlier.

The use of the Internet as a base for expert system development presents several special problems to developers. First, there is a need to cope with emerging technology, such as new versions of intelligent tools, servers, browsers, programming languages etc. The use of the Internet raises the significance of this problem because of the variety of components and languages involved in an Internet-based expert system (inference engine, knowledge base, client–server software and interface components) and the rapid pace at which Internet technology is changing. Second, there is a need to provide decentralized support and training for users. Making access to expert systems widely available also creates the need for widely distributed support. The Internet can be used as a training and support medium, however, and expert system technology has been used to create online tutorial and help-desk applications. Third, there are problems related to communications speed encountered with the use of multimedia in expert systems. HTML-based user interfaces allow the incorporation of a rich variety of graphic, audio and visual material, all of which require significant bandwidth for delivery. If users are restricted to the use of slower connections, or if many users access the system simultaneously, the communication requirements for multimedia components can create a bottleneck.

3. 2. Tools and languages available

Several tools and languages are available for developing intelligent Internet-based applications. In general, these tools employ traditional expert system techniques and offer in addition the capacity for Web-based development. While the list is not exhaustive, it is representative of tools available for Web-based expert system construction.

- Acquire, by Acquired Intelligence Inc., is an expert system shell that also includes a knowledge acquisition tool. Acquire's knowledge structures include qualitative influence graphs, decision tables and traditional rules. Inference can be both rule-based and pattern-based. The development of Internet-based user interfaces is supported through a client–server development kit that supports Java and ActiveX controls. Additional information, including demos, is available at http://www.aiinc.ca.
- ExSys by Multilogic is an expert system shell incorporating rule-based and fuzzy reasoning. An Internet-based runtime component supports client–server designs through CGI scripts and can integrate other server-side components. Additional technical information is available at http://www.multilogic.com/solutions/default.cfm.
- The Java Expert System Shell (Jess) is an expert system shell in Java that processes a CLIPS-like rule-based language. Jess includes backward chaining as well, and has many object-oriented features including a direct interface to Java components. Additional information is available at http://herzberg.ca.sandia.gov/jess.
- KB Agent by Explore Reasoning Systems is an expert system shell based on the SOAR system of Allen Newell. It includes a CORBA interface for Internet-based applications. More information and demos are available at http://www.ers.com.
- XpertRule KBS is a rule-based expert system shell that interfaces over the Web with a thin client using Microsoft's Active Server Page technology. More information is available at http://www.attar.com.

4. Case study — the Reptile Identification Helper (RIH)

The RIH is an Internet-based expert system that aids in the identification of amphibian and reptile specimens as part of an ongoing census of amphibians and reptiles in the state of Pennsylvania, known as the Pennsylvania Herpetological Atlas (PHA). The RIH makes expert herpetological advice available via the Internet for PHA workers who are attempting to identify specimens sighted in the field. The system employs a multimedia interface to lead workers through a deductive process that produces a list of likely specimens with associated likelihoods (Figure 1).

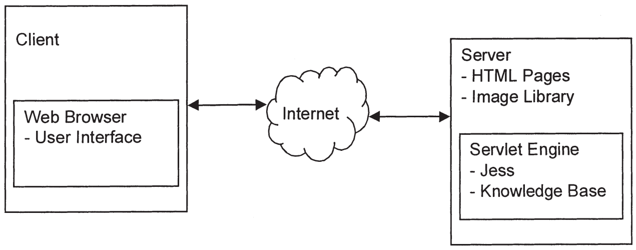

Figure 1: *RIH architecture.*

At the outset of the RIH project, a set of five design goals was identified, based upon the project objectives. The first goal was accessibility. To make it as widely available as possible, in homes, schools and offices, the system is distributed via the Internet and requires only common desktop software found on virtually all personal computers. The second goal was that the system should be self-evident. Since the user base is widely distributed and no training is provided for system users, it is necessary that first-time users with only basic computer skills be able to use the system successfully. The third goal was to employ a multimedia interface using text and graphics in order to support users' ability to make decisions and therefore increase the accuracy of identifications. The fourth goal involved the choice of a full-feature expert system shell, one that supports different modes of reasoning and that separates the knowledge base from interface programming, communications and inference portions of the system. Doing this allows the knowledge base to be extended and modified without requiring modifications to the underlying system. The final goal was to use freely available software and existing media, in order to minimize development costs.

To the user, the RIH appears as a guided interrogation supported by images that illustrate answer alternatives and a glossary to explain difficult terms. Identifying a specimen involves answering up to 10 questions concerning features of the specimen being identified or of its environment, choosing from two to five alternatives at each point. Figure 2, for example, illustrates a question concerning the type of scales of a snake. The word 'keeled' in this case is underlined, indicating that an explanation of the term is available to the user. Each alternative is illustrated by an image, which can be enlarged with a single click in order to see additional details. Figure 3 shows the same query, with the left image enlarged. Enlarged images disappear automatically upon the next mouse click. At the conclusion of the dialog, the RIH presents the user with a list of one to five candidate species to which the specimen being identified might belong, based upon expert opinion. Each species in the final list is ranked with a likelihood factor that indicates how strongly the expert feels that the user might actually be identifying a specimen of that species. Figure 4 shows the outcome of a session involving the identification of a turtle. The RIH has identified two candidate species in this case.

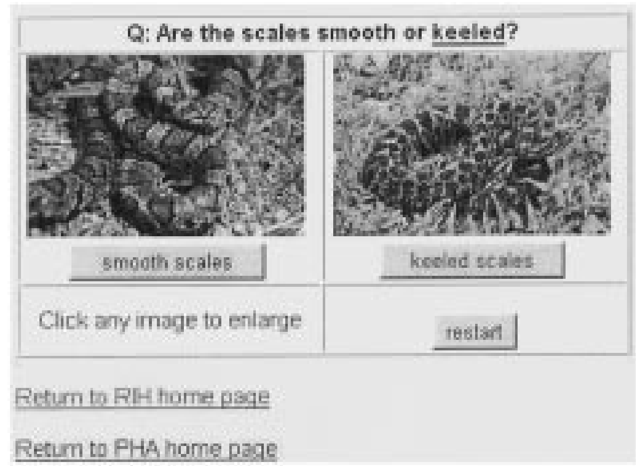

Figure 2: *User dialog — query.*

Figure 3: *User dialog — enlargement.*

Figure 4: *User dialog — results.*

The intelligent component of the RIH is based on the Java Expert System Shell (Jess) (Friedman-Hill, 1998). Jess was originally developed to support research into intelligent systems on the Internet. The Jess language, used to develop the expert system rule base, is a variant of the CLIPS programming language, which has a successful history as a base for expert system development. Jess supports both forward and backward reasoning and has the full features of a programming language, thus providing flexibility for future development of the RIH. Jess itself is written in Java, a language compatible with Internet tools and working environments.

The first version of the RIH to be developed (V0.1) was a proof-of-concept prototype. The system was limited in scope to the salamander family, including 27 separate species. This group was large enough to provide an interesting test of the system, but small enough to be manageable. The V0 software was implemented as a Java applet, a format intended for small pieces of software and not suitable for programs the size of the RIH. Initiating the V0 RIH took from 1 to 5 min, depending on the communication speed between the client and the Internet, clearly not practical for a working system. However, through prototype trials RIH V0 did provide proof that an expert system could be successful for this application.

RIH V1.4 is the current version. It is a true client–server system, in which the client software supports the user interface and the geographically separated server software supports the expert system. The Internet, of course, provides the client–server linkage. The client in this case is a Web browser, Netscape or Internet Explorer most probably, running on the user's computer anywhere in the state of Pennsylvania (though no firm geographic limitation exists). The server package includes a standard HTML-compatible Web server and a servlet engine. The Web server handles communications with the client and accesses the image library as necessary. The servlet engine supports the expert system shell. User inputs, which originate with the client software, are received by the Web server and are handed in turn to the expert system shell running under the servlet engine. Output generated by the expert system is handed to the Web server and then relayed via the Internet to the client. Currently the Web server is Microsoft IIS4.0 and the servlet engine is JRun from Allaire Corporation, though several comparable tools exist.

An RIH user requests service by pressing the START button on the RIH home page. This triggers a request, via the server, to the servlet engine for service from the RIH servlet. The servlet engine first creates a new session object corresponding to the incoming request and then invokes the requested servlet in a unique thread for that service request. The servlet then begins running its 'start' method in that thread. The 'start' method begins by instantiating several objects, including a controller, an inference engine and an output generator, all of which are bound to the session object. The controller initiates the inference engine, which in turn initiates the client–server dialog by signaling the output generator. The output consists of HTML code with embedded Javascript and references to the image library. The generated HTML pages also contain FORMS controls that trigger the RIH servlet again when pressed. A technique known as 'URL rewriting' maintains session state, so that objects that were bound to the session can be located upon subsequent contact by the client. This connects each subsequent client request within the same dialog to the same inference engine. After a predefined period of inactivity, the servlet engine cancels the session and all related objects (controller, inference engine and output generator) are disposed of.

The RIH system, exclusive of Jess, comprises about 800 lines of code in five classes and one HTML document. The Jess package includes an additional 16,000 lines of code in 168 classes. The RIH knowledge base includes an additional 1200 lines of rule-based coding. The most difficult part of the RIH coding was required in the output generator, where HTML code is generated to be returned to the client. One method, written in Java, outputs HTML code that contains embedded Javascript code, which generates new HTML code containing new embedded Javascript code. In that section of the program, the coding is nested five layers deep, in three different languages.

In operation, the initiation and response time for V1 is only a few seconds, much more practical than V0. After initial deployment of V1, it became apparent that the new bottleneck in system performance would be the downloading of images used to illustrate questions posed by the system. Image files were initially stored in GIF format, with sizes ranging from 20 to 110 kilobytes each. Downloading the largest images could require tens of seconds. This is not a large amount of time, but experience has shown that it is enough to annoy users and make the system impractical. Later the images were converted to JPEG format and were compressed, making the image files about one-fourth of their original size. A slight degradation of image quality did occur during the compression, but the compressed images are adequate for their intended purpose.

5. Summary

The Internet provides a convenient medium for the delivery of a variety of expert systems including applications in business, government, medicine and research. Use of the Internet for expert system development offers several advantages, including easy access, common interfaces and a variety of development tools. Problems associated with development include the need to keep up with rapid technological change, the need for decentralized support and the potential delivery bottleneck caused by communication loads. Common design patterns include traditional expert systems adapted from organic designs to Internet-based designs and highly intelligent agent software. Several commercial and freeware tools are available for Internet-based expert system development. Most display traditional expert system technology such as case-based and rule-based reasoning with additional features to support Internet development.

The familiarity of expert systems and their long record of commercial success have made the transition of expert system technology to the Internet inevitable. The projected growth of the Internet suggests that the development and adaptation of expert systems for the Internet will continue and will grow as well. The development of object brokerage protocols and information exchange languages has also made possible the development of cooperating expert systems and the integration of expert systems into other Internet applications. The future of expert systems might include applications that are more intelligent, interactive and mobile, which will blur the distinction between expert systems and agents. Another possibility is the offering of automated expertise by Application Service Providers on a fee per service basis. In any case, the viability and popularity of expert system technology appears to be enhanced, rather than threatened, by the Internet revolution.

References

Acquired Intelligence Inc. (accessed January 2000) Douglas Fir Cone and Seed Insects, http://www.for.gov.bc.ca/TIP/IID.

Bhattacharjee, S. (accessed January 2000) The Expert System for Thermodynamics, http://eng.sdsu.edu/testcenter/.

Charles, J. (1999) Interplanetary network aims for the stars, *Computer*, **32** (9), 16–18, 21.

Chizzali-Bonfadin, C., K.P. Adlassnig, M. Kreihsl, A. Hatvan and W. Horak (1997) A WWW-accessible knowledge base for the interpretation of hepatitis serologic tests, *International Journal of Medical Informatics*, **47** (1–2), 57–60.

Friedman-Hill, E. (1998) Java Expert System Shell (Jess), *Technical Report #SAND98–8206*, Sandia National Laboratories, Livermore, CA.

Grove, R.F. and A.C. Hulse (1999) An Internet-based expert system for reptile identification, *First International Conference on the Practical Application of Java, London*. Blackpool: Practical Application, 165–173.

Hewlett Packard Inc. (accessed January 2000) Configuration and Sizing Tool (CAST) for SAP BW, http://cast.hp.com/cast.

Kao, J.-J. (1998) Multimedia network system for landfill siting, *Proceedings of the Annual Water Resources Planning and Management Conference, Chicago, IL*. Reston, VA: American Society of Civil Engineers, 443–448.

af Klercker, T. and M. af Klercker (1998) Decision support system for primary health care in an Inter/Intranet environment, *Computer Methods and Programs in Biomedicine*, **55** (1), 31–37.

NUA Internet Surveys (accessed January 2000) Nua Internet How Many Online, http://www.nua.ie/surveys/how_many_online/index.html.

Research Triangle Institute (accessed January 2000) Coating Alternatives Guide, http://cage.rti.org.

Research Triangle Institute (accessed January 2000) Solvent Alternatives Guide, http://sage.rti.org.

Riva, A., R. Bellazzi and M. Stefanelli (1997) A Web-based system for the intelligent management of diabetic patients, *M.D. Computing*, **14** (5), 360–364.

Simpson, J., J. Kingston and N. Molony (1998) Internet-based decision support for evidence-based medicine, *Applications and Innovations in Expert Systems VI, Proceedings of expert system98, the Eighteenth Annual International Conference of the British Computer Society Specialist Group on Expert Systems, Cambridge*. London: Springer, 149–164.

Stern, E. (1999) The OSHA Hazard Awareness Advisor, *PC-AI*, **13** (2), 27–30.

Sycara, K.P. (1998) The many faces of agents, *AI Magazine*, **19** (2), 11–12.

Willard, K., J.H. Hallgren, B. Sielaff and D.P. Connelly (1995) The deployment of a World Wide Web based medical information system, *Symposium on Computer Applications in Medical Care*. Philadelphia, PA: Hanley & Belfus, 771–775.

The author

Ralph Grove

Ralph Grove is an Associate Professor of Computer Science at James Madison University in Harrisonburg, Virginia, USA. His background includes degrees in Computer Science from Purdue University and the University of Louisville, as well as experience in business information systems development and management. Research interests include software engineering, artificial intelligence, and computing and the environment.