# Design and Development of Knowledge-Based Systems on the Web

**Ralph F. Grove**
**Indiana University of Pennsylvania**
**Computer Science Department**
**Indiana, PA 15705**
**rfgrove@computer.org**

**Abstract**

The World Wide Web (Web) offers a large potential for delivery of various information-based services, including the services of intelligent applications. As access to the Web has grown its value as a medium for the delivery of knowledge based systems (KBS), in particular, has increased. There are now a large number of KBS available on the Web and a variety of development tools to support Web-based KBS development. This paper discusses several issues relating to Web based KBS applications, including development tools and design issues.

**Keywords:** Knowledge Based Systems, Expert Systems, Intelligent Agents, Internet, World Wide Web

## Introduction

The first wave of Knowledge Based System (KBS) applications, developed in the 1970's and 80's, were stand-alone applications, based on mainframe or PC platforms, or LAN-based distributed applications. Classical expert systems include MYCIN - an expert for diagnosing blood disorders, Prospector - for mineral exploration, Dendral - for determining molecular structure of organic molecules, and R1/XCON - for configuring computer systems. Jackson gives a comprehensive introduction to expert system technology in his book Expert Systems [3].

Several problems have been associated with traditional KBS applications. The *Knowledge Bottleneck* relates to the difficulty of acquiring knowledge during system development, and of maintaining knowledge bases over time. Contributing factors include the limited availability of expertise, the difficulty of verifying encoded knowledge, and the dynamic nature of knowledge. KBS applications often exhibit *brittleness*, i.e., they have a limited area of expertise and perform poorly outside of that boundary. Furthermore, there is no inherent awareness of this limit in most KBS. Another problem is *availability*, i.e., having the expertise provided by a KBS at the place and time where it is needed. A KBS having an interface that is located in a physician's office will be of little use to the physician working under pressure at a patient's bedside. For many users, even a slight inconvenience is enough of an obstacle to acceptance to cause the failure of KBS development. *software distribution* can be a problem for expert system developers. 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. The issue of *communication* is problematic for distributed applications. A lack of common protocols for the exchange of knowledge among KBS has tended to discourage designs involving cooperation or dynamic information sharing in applications.

There are several factors that combine to make the Internet, by contrast to stand-alone platforms, an ideal base for KBS delivery. *The Internet is widely accessible*. Internet usage continues to grow rapidly, with varying estimates putting the number of Internet users at over 100 million in the US and up to 300 million worldwide. In addition, business Intranets provide access in many workplaces. *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 multimedia user interface platform on which developers can build. *Several Internet-compatible tools for KBS development are available.* There are a variety of KBS development tools that support internet-based interfaces including both commercial and freeware development environments. *Internet-based applications are inherently portable*. The Internet provides a development environment that is platform independent and ubiquitous. This means that there is no need for special distribution and installation of KBS software in advance of its use. Rather, applications are provided on demand at the time and place they are needed. *Emerging protocols support cooperation among KBS.* CORBA and XML in particular provide common protocols for the exchange of information and services, which could be used to develop cooperation among KBS as well.

There are now a number of KBS available on the Web, representing industrial, medical, scientific, and academic application domains. There are applications, for example, to choose an appropriate coating for manufactured parts, to solve an equation in thermodynamics, to identify a biological specimen, and to identify workplace hazards. Links to a sample of publicly available applications can be found at http://www.co.iup.edu/~rfgrove/webbasedKBS.

**Design Issues**

Several design patterns have emerged from efforts in Web-based KBS development and research. *Web-based expert systems* are traditional expert systems, using mainly rule-based and case-based reasoning, that have been adapted from organic designs to Internet use by incorporating client-server architectures and Web-based interfaces. *Intranet-based expert systems* are similar, except that access is restricted to a local corporate setting for security, privacy, or legal reasons. In medical applications, for example, there is a requirement for strict privacy and protection of patient records. *Cooperative Expert Systems* are expert systems that can jointly solve a problem by sharing expertise to synthesize a solution. Web protocols such as XML and CORBA provide the communication medium through which this cooperation can occur. One objective of research into this design is to overcome the limitation of expertise characteristic of most KBS. *Intelligent Agents* - Alternatively referred to as softbots or knowbots, highly intelligent applications are now commonly known as *agents*. Though the term lacks an accepted and precise definition, it is generally reserved for applications that have some degree of reactivity, autonomy, and adaptability [4]. 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.

Several tools and languages are available for developing web based KBS applications. Generally these tools use conventional KBS reasoning modes such rule-based inference, decision-tree rule induction, Bayesian reasoning, fuzzy logic, etc., and are tightly integrated with Web technologies for managing the user interface and client-server components of the application. Typical of these are: Java Expert System Shell (Jess), a CLIPS-like rule-based language that includes object-oriented features and direct interface to Java objects. KB Agent by Explore Reasoning Systems is an expert-system shell based on the SOAR package. It incorporates a CORBA interface and C++ language API. ExSys by Multilogic is an expert system shell incorporating rule-based and fuzzy reasoning. The Web interface is provided by CGI/Perl components in this case. The XPertRule KBS shell performs induction of rules based on decision trees, and incorporates fuzzy logic as well. It interfaces to the Web via Active Server Pages or via a custom thin client component. Links to a variety of these tools can be found at http://www.co.iup.edu/~rfgrove/webbasedKBS.

The use of the Internet as a base for KBS 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. This use of the Internet raises the significance of this problem due to the variety of components and languages required to support Internet-based KBS. *Second*, there is a need to provide decentralized support and training for users. Making access to KBS widely available also creates the need for widely distributed support. The Internet can also be used as a training and support medium, however, and KBS technology can be used to create online tutorial and help-desk applications. *Third*, there are problems related to communications speed encountered with the use of multimedia in KBS. 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 can create a bottleneck in the system.

**Case Study - The Reptile Identification Helper**

The Reptile Identification Helper (RIH) [2] is a Web-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 likelihood factors. The RIH is available on the Web at http://www.nsm.iup.edu/pha/rih.

At the outset of the RIH project, a set of five design goals was identified, based upon the project objectives described in the introduction. 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, images, 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 which supports different modes of reasoning and which separates the knowledge base from programming of the interface, communications, and inference portions of the system. Doing so 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. A typical RIH session involves answering about 10 questions, choosing from two to five alternatives at each point. The questions concern features of the specimen being identified or of its environment. Each alternative is illustrated by an image, which can be enlarged with a single click in order to see additional details. Figure 1 shows a typical query.

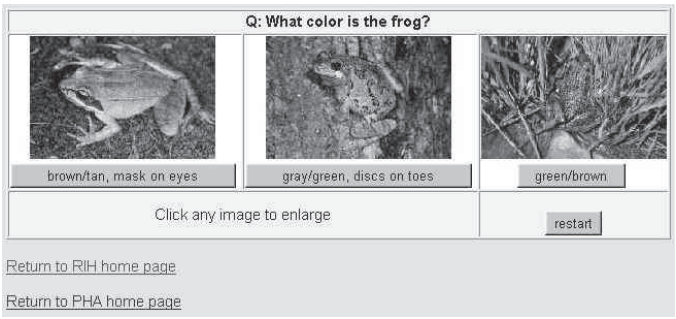


Figure 1. Typical Query Panel

At the conclusion of the dialog, the RIH presents the user with a list (one to five) of possible 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 be actually identifying a specimen of that species. The likelihood factors are determined in advance by the domain expert and coded into the knowledge base. Figure 2 shows a typical result from an RIH session.

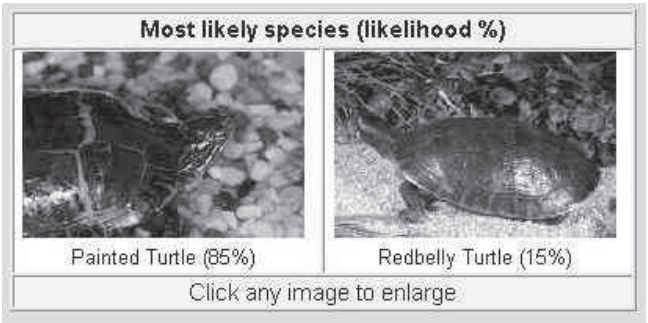


Figure 2. Typical Result Panel

The intelligent component of the RIH is based on the Java Expert System Shell (Jess) [1]. Jess was developed as a tool 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 NASA CLIPS programming language, which has a successful history as a base for expert system development. Jess supports both forward and backward reasoning and interfaces directly to Java objects, giving it the full features of the Java 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.

RIH V.1 is the current version of the system. 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 likely, 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 interface engine. User inputs, which originate with the client software, are received by the Web server and are handed in turn to the inference engine 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 LiveSoftware, though several comparable tools exist.

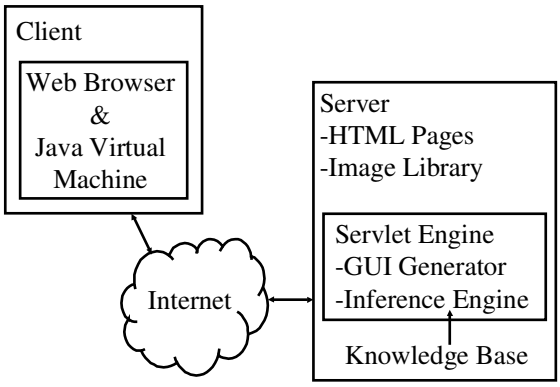


Figure 3. RIH Architecture

**Summary**

The World Wide Web has provided a convenient medium for the delivery of a variety of KBS including expert systems and intelligent agents. A variety of knowledge-based applications have appeared on the Web already, including applications in education, medicine, science, and industry. Use of the Web for KBS development offers several advantages over traditional stand-alone or LAN based applications, including widespread access, common user interfaces, a variety of development tools, and evolving communication protocols to support interaction among intelligent applications.

Several problems are associated with development of Web based KBS. One is the need to keep up with rapid technological change, including servers, interface components, inference engines, and various protocols. Another is the need for decentralized support, as the user base grows in size and in geographical distribution. A third is the potential delivery bottleneck caused by communication loads and limited infrastructure.

Several architectural paradigms have appeared in development and research into Web based KBS. Traditional expert system technology has been adapted from organic designs to Web based applications through the incorporation of Web interfaces. Intranet based applications have appeared in domains where privacy and security are paramount. Research into cooperative expert systems seeks to overcome the limitations and brittleness of standalone expert systems. Research into Intelligent agents seeks to develop complex, autonomous, and adaptable intelligent applications.

In addition to the problems mention above, one open question at this point is the issue of design - how best to integrate KBS services with a user interface using Web based technologies. This is a very dynamic problem at the moment, due to the constant change in Web technologies. Another issue is that of communication and cooperation among KBS. Specific problems include maintaining directories of available services, cost models, and establishing common protocols for system interaction and for domain level knowledge sharing.

Projected growth of the Internet suggests that the development and adaptation of Web based KBS will continue. As the Internet expands into new domains, both personal and commercial, and as evolving technology expands the capabilities of KBS, Web based experts and agents will become more sophisticated, complex, and capable.

**References**

[1] Friedman-Hill, Ernest. "Java Expert System Shell (Jess)". Technical Report #SAND98-8206, Sandia National Labs, Livermore, CA, 1998.

[2] Grove, Ralph F., Arthur C. Hulse. "An Internet-Based Expert System for Reptile Identification". The First International Conference on the Practical Application of Java, London, UK, pp. 165-173, April 23 1999

[3] Jackson, Peter. Introduction to Expert Systems. 3rd ed., Addison-Wesley, Harlow, England, 1999.

[4] Sycara, Katia P. "The Many Faces of Agents". AI Magazine, 19/2:11-12, 1998.