

Ex-W-Pert System: A WWW-Based Distributed Expert System

—WWW 上での分散型エキスパートシステムの実現方法—

Behrouz H. Far * and Zenya Koono

Department of Information and Computer Sciences, Saitama University, Japan
埼玉大学 情報システム工学科

Abstract

The World Wide Web (WWW) allows people at remotely located sites to communicate and share their ideas using a common communication protocol. A common use of the WWW system is running a client application, using a browsing tool, by pointing to a local or proxy server to browse files written in the hypertext format that contain anchors addressing other URLs. In this paper a new application of the WWW in a distributed knowledge based system is introduced and applied to a teamwork design. First, an architecture for a WWW-based Distributed Expert System (Ex-W-Pert System) is proposed and, second, an implementation of the proposed structure in groupware design is demonstrated. The resources and knowledge bases are distributed and can be accessed through the internet.

1 Introduction

The World Wide Web (WWW) allows people at remotely located sites to communicate and share their ideas using a common communication protocol that can handle text files, images, sounds, forms, etc. [8]. Using WWW, one can address a file by simply calling its URL (Universal Resource Locator)[12], which is composed of the protocol name (http, ftp, gopher, etc.), host, domain and path, respectively. HTTP is an internet protocol which runs over a TCP connection in order to transfer data and hypertext files written in hypertext mark up language (HTML)[1].

Presently, a common use of WWW system is running a client application, using a browsing tool, pointing to a server to browse hypertext documents[9]. The hypertext documents contain anchors that link them to other URLs, allowing connection to other servers.

In this paper we introduce a new application of the WWW system in sharing knowledge bases together with the other data items. The research is pointed towards building distributed expert systems with physically distributed knowledge bases that can be shared over the internet. The main idea is using a *communication engine* together with a conventional *inference engine* and *knowledge-base* in an expert system.

This is directly applied to teamwork development activities such as design. We introduce an architecture for a WWW-based Distributed Expert System (Ex-W-Pert System)

*Contact address:

〒338 埼玉県 浦和市 下大久保 255 埼玉大学 情報システム工学科

Tel: +81-48-858-3489 Fax: +81-48-858-3716

E-mail: far@cit.ics.saitama-u.ac.jp WWW: <http://www.cit.ics.saitama-u.ac.jp/~far/>

and an implementation of a groupware design system that resembles a team of human designers assigned to cooperatively design an artifact.

Applied AI systems have long been suffering from the lack of efficient data search and retrieval in the large scale knowledge bases. We think that a WWW based distributed knowledge base and efficient data communication is a possible solution. This paper is a first step towards realizing this idea by blending the network communication and expert system technology.

2 Ex-W-Pert System

Figure 1 shows an overview of the Ex-W-Pert system architecture that is used as a test bed. This system is composed of engineering workstations and personal computers that run their own local expert system client. Each platform with a running expert system client is called an *expert unit*. Local expert units are connected by the Local Area Network (LAN) and communicate using the NFS and HTTP protocols.

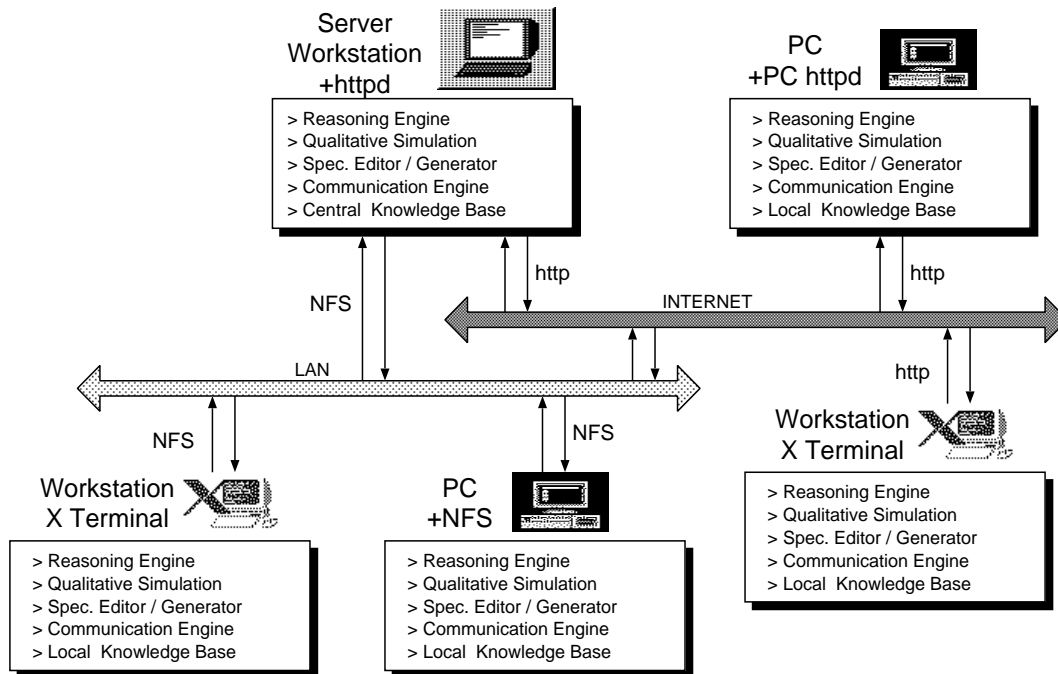


Figure 1: Overview of the Ex-W-Pert System

Remotely located expert units have access to the internet and communicate with other units through the HTTP protocol. All the local units need not run a HTTP daemon if there is a HTTP server workstation supporting them all. However, remote units serving data and documents, must be equipped with a running HTTP server daemon. Free and commercial HTTP server daemons are available for Unix, Mac and Windows platforms.

Similar to conventional systems, each expert system unit, no matter local or remote, has its own *local knowledge-base* and *reasoning engine*. Compared to the conventional expert systems, a main difference is that all expert units have an additional *communication engine*.

This architecture is general enough to implement any kind of groupware expert system. In this paper an implementation of the Ex-W-Pert system in groupware design is demonstrated.

3 Groupware Design Application

Ex-W-Pert system is partially implemented in the experimental groupware design system, focusing on design verification by assisting a team of human designers through blending the knowledge acquisition, documentation and reasoning procedure, using local and remote knowledge bases.

Ex-W-Pert system is composed of a number of expert units (see Fig. 1). All the units have a common architecture as shown in Fig. 2. This is an enhanced version of our standard architecture of expert systems used for design[7].

Each expert unit is composed of:

- A data translator for converting specification and design input to an internal frame data structure and vice-versa.
- A reasoning engine together with a qualitative simulator, customization and learning modules for reasoning and simulating the behavior of the design artifact.
- A local knowledge base, including a domain-oriented library of component models, design procedures and design documents.
- A window-based user interface that allows a user interactively select arrangement of components, view simulation results, view and edit specification and documents.
- A design documentation module for recording the partial results and making hyper-text design documents.
- A communication engine, for launching http, ftp, etc., applications and communicating to the other units.

The data translation, reasoning and data communication modules are briefly described in the following subsections.

The system is partially implemented using Common Lisp, C, TCL/TK and Perl. Common Lisp and C are used to implement the reasoning and simulation modules. TCL/TK is used to implement the user interface and basic communication methods. Perl is used to add additional features to the communication methods.

3.1 Data translation module

A designer prepares an initial design sketch using defined components. Simple graphic symbols are provided to ease this step. The initial design input is converted to a frame structure, using the components library of the local knowledge base, suitable for processing by the reasoning module. The `#INPUT_TO_FRAME` module is used for parsing and converting the design input to the frame structure suitable for processing by the reasoning module. Presently, a set of pictorial elements are used to represent some frequently used components in chemical process design, such as valve, tank, etc. Another set is developed for software design. This includes a subset of symbols used in Specification and Description Language (SDL) [4]. Yet other sets can be developed for other applications, if required.

There is also a set of design goals that must be fulfilled by this design sketch. The goals are derived by analyzing the input specification. We use a kind of natural language with restricted syntax and semantics to reformat the specification[11]. Sentences in this restricted syntax language are distinguished by their type, subject, condition(s), verb, etc. The following sentence is an example:

#type S(subject) V(verb) E(condition[s]);

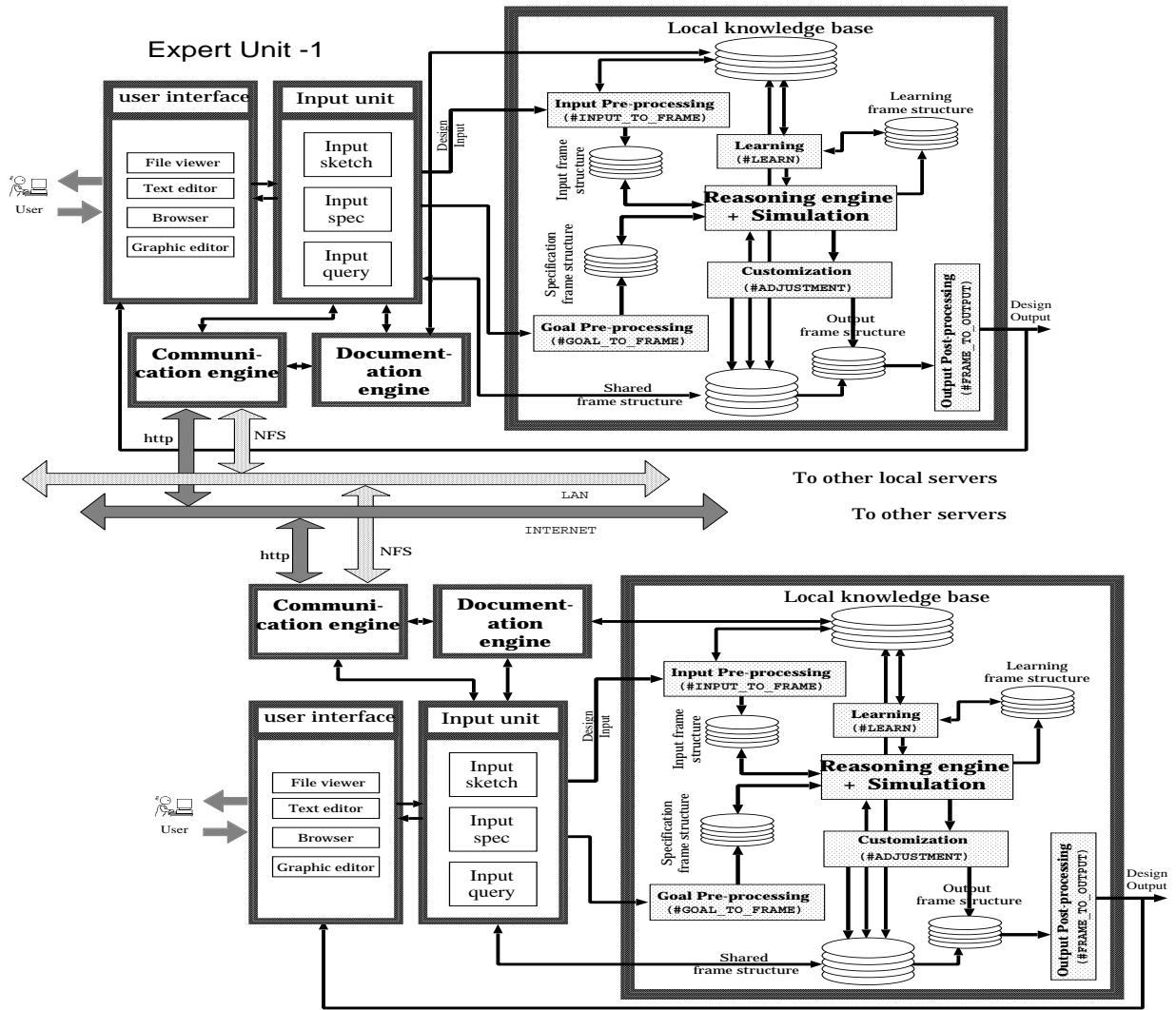


Figure 2: Experimental groupware expert system

Here (#) represents the type, S(.) is subject, V(.) is verb and E(.) is condition(s). This is useful for efficient parsing and reducing the ambiguity of ordinary natural language sentences.

Then the #GOAL_TO_FRAME module converts the spec to an internal frame structure that is used by the reasoning engine later.

Also, the results detailing and customization are recorded in the created frame structure which is finally converted to a final design sketch using the #FRAME_TO_OUTPUT module. The designer can check and modify the results, interactively.

3.2 Reasoning module

In groupware design, the human experts communicate using a language that is built around the *function* and *behavior* of the components that are used in the design. They mainly talk about the contribution of a component to the functionality of the whole design and how the system may behave if a component is added to it or removed from it. This language has its roots in the components' physical model but uses a syntax different from the direct physical and geometric models.

In this project, Functional Reasoning (FR) is used for deriving and explaining function

of a device from a qualitative description of its structure and behavior through systematic generation and reasoning with such a model at various levels of abstraction [5, 6].

A function concept is defined as an interpretation of a persistence or an order in the sequence of qualitative states, using trace of qualitative state vector derived by simulation on a qualitative model of a device. The qualitative model is composed of a set of expressions involving three primitives: qualitative variables, ordinary- and coordinative- qualitative operations. Qualitative variables are counterpart of physical quantities, such as temperature and pressure, representing characteristics of the system's inner environment. Relation between the qualitative variables is defined by qualitative operations. Ordinary operations are monotonic increase (M^+), monotonic decrease (M^-), positive influence (I^+) and negative influence (I^-). Coordinative operations account for protocol-based interactions, such as 'when', 'until', 'set', 'reset', and 'default'.

Behavioral Fragment (BF) is defined as the record of landmark values of the qualitative variables. BFs are derived by qualitative simulation in two steps: [6] (i) dependency constraint satisfaction; and (ii) landmark value identification of the qualitative variables. BFs are sequences leading to a function concept.

A *Function* concept can be derived if a repetition cycle or an order (e.g. persistence, etc.) can be detected on a BF sequences. The repetition cycle or persistence are derived for each of the qualitative variables and different cycles can possibly be detected. Each cycle represents a function concept from a different viewpoint. The function derived is a direct consequence of the designer interacting with the qualitative representation of the underlying physical model of the device and physical constraints [6]. This is a very useful characteristic of the technique and can be used directly in functional design. Particularly, by recording the goal(s) of the designer on each step and comparing it with the derived function, one can easily verify whether the arrangement can satisfy the intended goal(s) of the designer.

There are already two set of frames for the input file and design goals. The reasoning engine is responsible for checking the input frame structure, fetching appropriate components, acquiring their function and examining the functions against the design goals. The components that satisfy design goals are recorded in the output frame structure. This is done repetitively until all design goals are satisfied.

The #LEARN module keeps record of the design steps. This is necessary for saving resources in similar design cases. The result of matching is then delivered to the #ADJUSTMENT module that is responsible for customizing the candidate components.

3.3 Local knowledge-base

The local knowledge-base is composed of a domain-oriented library of the components model, recorded design procedure and design documents. A main advantage of the Ex-W-Pert system is that documents, procedures and component models related to a particular design task can be focused in a particular expert unit and gradually this unit will grow up to have expertise in that particular area. All the other units requiring such expertise acquire its service using HTTP communication engine.

3.4 Communication module

The communication module is the heart of the system. The communication module has three main functions:

- Acquiring data from the other sites, as requested by the reasoning engine.

- Performing index and relational keyword search.
- Preparing and reformatting data items to be appropriate for transferring over the internet.

All the expert units understand and communicate through the HTTP. Each unit has a local index of its own knowledge items and a moderate index of knowledge items on the other sites. A comprehensive index resides in one of the units that is assigned to act as the main index search server and offers such a service to all the unit members.

Figure 3 shows an example data acquisition handled by the data communication unit.

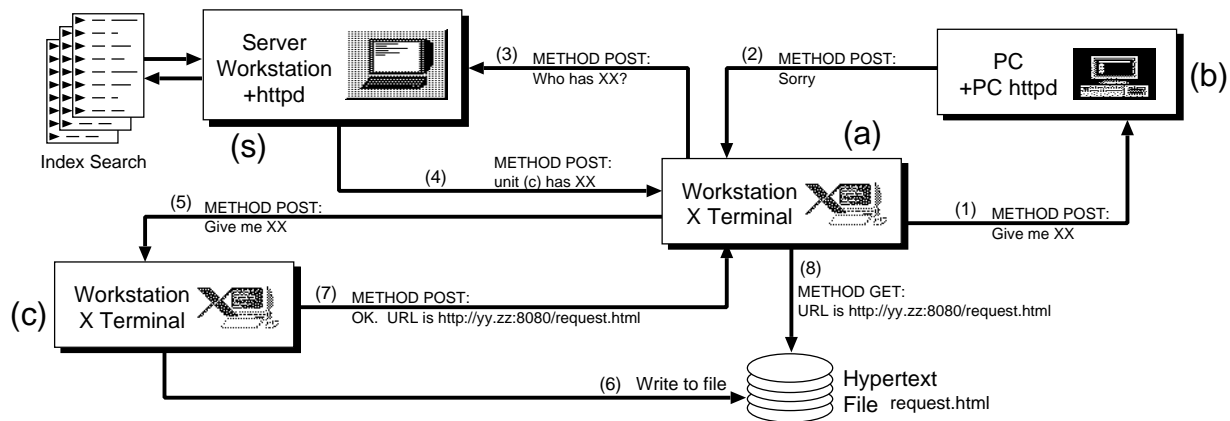


Figure 3: A communication example

In *step 1* a request for additional information from any unit, say unit (a), activates a POST method[9] and asks for information from a unit that most likely has such information. Such a request is first acknowledged by recipient machine (b) that seems to have knowledge related to the current query. The local index of unit (a) is updated every time a new POST is acknowledged.

If this request fails, i.e., unit (b) does not acknowledge the request, as in *step 2*, the same query will be posted to the server workstation (s) that maintains a general index of the data items in the remote knowledge bases (*step 3*). In *step 4*, the most appropriate site for the requested information is selected. This is announced back to the unit (a) with the name of the unit that has such information, in this case (c). Again in *step 5* unit (a) launches a request aiming at unit (c) using method POST. Unit (c) copies the requested information in a file (*step 6*) and gives the URL of the requested information to unit (a) (*step 7*). Then a GET method by unit (a) is launched to fetch the URL. This URL is a hypertext file that has some text entries and some other data, such as image, simulation results, etc. The data entries are parsed and added to the knowledge base of unit (a). In some occasional cases where the network traffic is high and the two sites allow remotely writable memory areas, a PUT method, say by site (c) may be launched to write the requested file into the disk of unit (a).

Communication through the HTTP requires all the transferable information be first written into a hypertext file and then transmitted. Therefore the data communication module is responsible for preparing the information and rewriting the information in the hypertext form before delivering it to the WWW system.

In conventional application of the WWW system running the GET, POST and PUT methods and data preparation tasks requires launching a client software. In the Ex-W-Pert system we have implemented programs that can run the methods and reformat the

hypertext data automatically without using a client software. Another advantage is that all the communication activities are done at the background and the user is not required to be aware of the HTTP connections. Therefore, it is not necessary for the user to have networking and WWW knowledge in order to use and interact with the system.

3.5 User interface module

No matter what the platform is, all the units have the same user interface and the users view the same window. The user interface module for various platforms, using TCL/TK toolkit, is currently under construction.

4 Discussion

4.1 Other works

The idea of *Groupware Systems*[3] has been around for a while, considering computer support of a team work. There are already some examples of groupware applications such as bulletin boards, cooperative games, video conferencing system and screen sharing cooperative design systems[10]. Most of these systems concentrate on techniques to integrate sound and video communication over a network of clients using the new features offered by broad-band digital network (B-ISDN).

Most of the work on distributed AI systems have concentrated on the reasoning and conflict resolution methods[2]. A common assumption in configuring the structure of such systems is that they consider clients connected through a kind of local area network (LAN). This limits the scope of application of such systems drastically. To the best of our knowledge, there is no live project seeking integration of distributed expert systems and knowledge bases assuming its clients are distributed over the internet. In our project various hardware (PC, Workstation, Mac, etc.) with various operating systems (MAC, WINDOWS, UNIX, etc.) and various methods of connection to the internet (LAN, ISDN, modem with PPP account, etc.) are considered and a mechanism for communication and cooperation is proposed.

4.2 Network considerations

Besides the experimental methodologies, there are three generally accepted and wide spread choices to implement a distributed architecture on the internet, i.e., *WAIS*, *Gopher*, and *HTTP*. Among these, only *HTTP* is targeted and designed to fully functionalize a collaborative work space. *WAIS* allows fast text based data base search, using master and local indexes. However, it does not allow reference to other servers. *Gopher* allows text based search using menus, that can address other servers, in turn. The main method used in *Gopher* is *GET* for retrieving a document. *HTTP* allows using hypertext besides the menus. In other words, it offers methods for *GET*, *PUT* and *POST*. Using the latter methods is crucial to our project. A disadvantage of *HTTP* is its high cost and inefficient search. In our project, a platform running *HTTP* daemon talks to a *WAIS* server and as mentioned earlier, we use *WAIS* index search together with the *HTTP* for efficient searching of the knowledge bases. Another disadvantage of using HTTP is that the methods are usually launched by a browser client. In many cases two knowledge based systems that communicate and exchange information, launching a client software is unwanted and

brings unnecessary overhead. We have developed programs to enhance the functionality of the WWW methods without using a client.

5 Conclusion

In this paper, a new application of the World Wide Web technology in expert systems was discussed. An architecture for a WWW-based Distributed Expert System (Ex-W-Pert System) was proposed and a preliminary implementation of the proposed structure in groupware design was introduced. This system resembles design by a team of human experts. The resources and knowledge bases are distributed and can be accessed through the internet.

The idea of the Ex-W-Pert system is general and can be used to implement systems that communicate and acquire their knowledge from distributed knowledge bases systems. Currently, we are conducting further research on intelligent goal oriented search algorithm and an expert system shell with network connection capabilities is currently under development.

References

- [1] T. Berners-Lee, R. Cailliau, H.F. Nielsen and a. Secret, "The World-Wide Web," *Communications of The ACM*, vol. 37, no.8, pp. 76-82 (1994).
- [2] A.H. Bond and L. Gasser, "An Analysis of Problems and Research in DAI," *Readings in distributed Artificial Intelligence*, A.H. Bond and L. Gasser (edts.), Morgan Kaufmann Pub. Inc., pp. 3-35 (1988).
- [3] C.A. Ellis, S.J. Gibbs and G.L. Rein, "Groupware: Some Issues and Experiments," *Communications of The ACM*, vol. 34, no.1, pp. 38-58 (1991).
- [4] B.H. Far, T. Takizawa and Z. Koono, "Software Creation: An SDL-Based Expert System for Automatic Software Design," *SDL '93: Using Objects*, O. Færgemand and A. Sarma, eds., pp. 399-410, Elsevier Publishing Co., North-Holland, 1993.
- [5] B.H. Far and Z. Koono, "Integrating Device Modeling and Design Process Knowledge in Functional Design," *Proc. 11th Design Symposium, Japan Mechanical Engineering Society*, pp. 40-49, Tokyo, Japan, June, 1993.
- [6] B.H. Far and Z. Koono, "Functional Reasoning, Explanation and Analysis: A Collective View and A Proposal," *Proc. IJCAI' 93 Workshop on Engineering Problems for Qualitative Reasoning*, Chambery, France, August 1993, pp. 23-30.
- [7] B.H. Far, T. Takizawa and Z. Koono, "Software Creation: An Expert System for Reproducing Human Cognitive Processes in Automatic Software Design," in *Proc. World Congress on Expert Systems' 94*, Estoril, Lisbon, Portugal, (Jan 1994).
- [8] 'http://www.cit.ics.saitama-u.ac.jp/~far/Lectures/html_lecture/japanese/course-j.html'
- [9] '<http://www.w3.org/hypertext/WWW/Protocols/HTTP/HTTP2.html>'
- [10] H. Ishii, M. Kobayashi and K. Arita, "Interactive Design of Seamless Collaboration Media," *Communications of The ACM*, vol. 37, no.8, pp. 83-97 (1994).
- [11] T. Sugiyama, "ソフトウェアクリエーション：データ定義表から設計知識獲得," (Acquiring Design Knowledge from Data Definition Tables) (in Japanese), Report ICS-95B-12, Department of Information and Computer Sciences, Saitama University, (1995).
- [12] '<http://www.w3.org/hypertext/WWW/Addressing/Addressing.html>'