社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技法
TECHNICAL REPORT OF IEICE.

ソフトウェア設計用分散型エキスパートシステム

ベルーズ H. ファー†　河野 善彌

†〒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. 048-858-9612,　Fax. 048-858-3716
E-mail. far@cit.ics.saitama-u.ac.jp
http://www.cit.ics.saitama-u.ac.jp/~far/

あらまし:　本研究の目的は, ネットワーク技術を設計知識ベースと設計ドキュメントに適用することである。 散在する設計チームを対象とするには, 分散型知識ベース, 分散型設計ドキュメンテーション, 推論エンジンを実現しなければならない. 本システムの分散型知識ベースは、設計工程知識と設計対象知識を格納している. 分散型推論エンジンは, 機能推論とサーチ機能を持つ. ネットワーク上に複数存在する分散型推論エンジンは、各々が所有する知識ベースだけでなく他の知識ベースもインターネット経由で利用し、推論を進める ことができる. 分散型設計ドキュメンテーションは WWW を利用して設計ドキュメントを提供する. これにより各設計者が持っている設計知識を交換するだけでなく, 推論エンジン の支援によりあたかも他の設計者の知識を自己の知識のように活用できる環 境を実現できる. 従って, 各設計者が固有に持つ知識をグループ全体で利用で きるため, グループ全体の設計効率と質を高められる。

和文キーワード　*WWW*, **エキスパートシステム**, **グループウェア**, **設計**

A Distributed Expert System for Groupware Design

Behrouz H. FAR†　Zenya KOONO

†Department of Information and Computer Science,
Faculty of Engineering, Saitama University
255 Shimo-okubo, Urawa 338, Saitama, Japan
Tel. +81-48-858-9612,　Fax. +81-48-858-3716
E-mail. far@cit.ics.saitama-u.ac.jp
http://www.cit.ics.saitama-u.ac.jp/~far/

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

Key words　*WWW, Expert System, Groupware System, Design*

1 Introduction

The World Wide Web (Web) 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. [14]. What makes Web more attractive and useful is the capability of handling URLs (Universal Resource Locator) [13]. Using Web, one can address a file by simply calling the protocol (*HTTP*, *FTP*, gopher, etc.), host, access port and path, respectively. Presently, a common use of Web system is running a client application, using a browsing tool, pointing at a local or proxy server [12] to browse data written in the hypertext format [9]. The hypertext files contain anchors that address other URLs, and connection to other servers is possible.

In this paper we introduce an application of the Web in expert systems with shared and distributed knowledge bases. This is directly applied to teamwork development activities such as design. We introduce an architecture for a Web based Distributed Expert System (Ex-W-Pert System) 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 the large scale knowledge bases. Data search and retrieval in such knowledge bases is a bottleneck. We think that a Web based distributed knowledge base and efficient data communication is a possible solution. This paper is a first step towards realizing this idea. We think that this will become a milestone in developing expert systems and opens a new line of research and development by blending the network and expert system technology.

2 EX-W-PERT System Test-bed

Figure 1 shows an overview of the Ex-W-Pert system test-bed. This system is composed of various platforms (workstation, PC, etc.) that run their own local expert system client. Each platform with a running expert system client is called an *expert unit*. No matter what the platform is, all the units have the same user interface and the users view the same window. Local units are connected by the Local Area Network (LAN) and communicate using the *NFS* and *HTTP* protocols. *HTTP* is an internet protocol which runs over a TCP connection in order to transfer hypertext information [1].

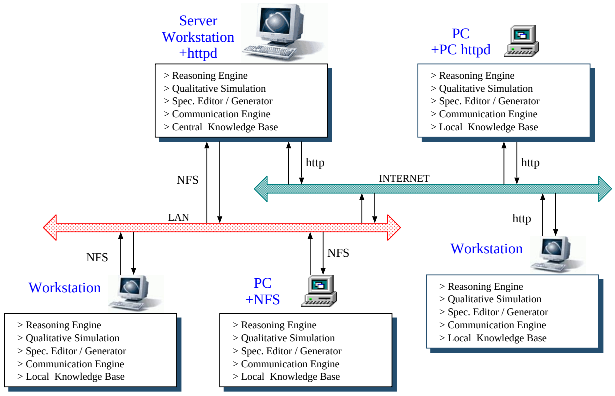

Figure 1: Overview of the Ex-W-Pert System test-bed.

Remotely located units have access to the internet. This is possible in many different way such as using modem and SLIP (or PPP) accounts, T1 connection, etc. Remote hosts communicate with the other units through the *HTTP*. All the local units need not run a *HTTP* daemon if there is only a server station that supports them all. However, for the remote units running *HTTP* daemon is a necessity, unless the proxy connection is allowed.

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 EX-W-PERT System for Groupware Design

Ex-W-Pert system is 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.

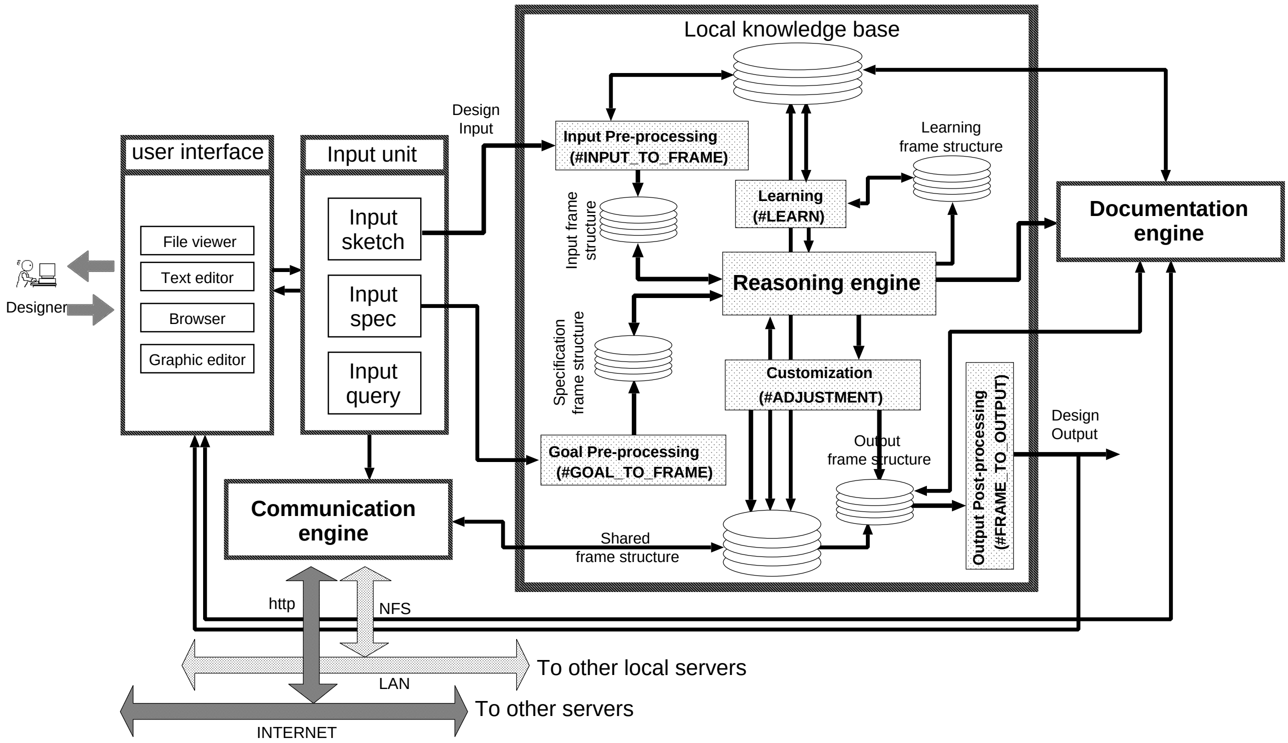

Figure 2: Experimental Ex-W-Pert groupware expert system.

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 for design [7].

Each expert system unit is composed of:

- A data translator for converting specified device model to frame data structure [7] and vice-versa.
- A reasoning module composed of an implementation of Qualitative Function Formation (QFF) algorithm [6] together with a qualitative simulator, customization and learning modules.
- Local knowledge base, including a domain-oriented library of component models and other design documents.
- A window-based user interface that allows a user interactively select arrangement of components, run simulation program, view simulation results, view and edit specification and documents.
- A design documentation module for recording the partial results and making hypertext design documents (optional).
- A communication engine, for launching *HTTP*, *FTP*, etc. applications and communicating to the other units.

The system is implemented using Common Lisp, C and Perl. Common Lisp and C are used to implement the reasoning and matching modules. C is used to implement the basic communication methods. Perl is used to add additional features to the communication methods.

The data translation, reasoning and communication engines are briefly described in the following sections.

3.1 Data translation module

A designer prepares an initial design sketch using defined components. Simple graphic symbols are designed to ease this step. The initial design input is converted to a frame structure, using the components library of the local knowledge base. 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 designed to represent some frequently used components in chemical process industry, such as valve, tank, etc. This set can be extended if required.

There is also a set of design goals that must be fulfilled by this design. 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. Sentences in the 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]*);

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 a frame structure that is used by the reasoning module 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 design and how the system may behave if a component is added to or removed from it. This language has its root in the components' physical model but uses a syntax different from the direct physical and geometric models.

In this project, Functional Reasoning (FR) technique 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. Qualitative Function Formation (QFF) technique [4; 5; 6] is developed and implemented in this system.

In QFF, 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 that we use in QFF 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^-) [11], positive influence (I^+) and negative influence (I^-) [8]. Coordinative operations account for interactive or protocol-based interactions, such as *'when'*, *'until'*, *'set'*, *'reset'*, and *'default'*.

Behavioral Fragment (BF) [4] 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 by QFF 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.

The Specification matching engine `#SME` module stays as the core of the system. There are already two set of frames for the input file and design goals. The `#SME` module is responsible for checking the input frame structure, preparing component pairs, acquiring their function from the `#QFF` 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 time in similar design cases. The result of matching is then delivered to the `#ADJUSTMENT` module that is responsible for customizing the candidate components.

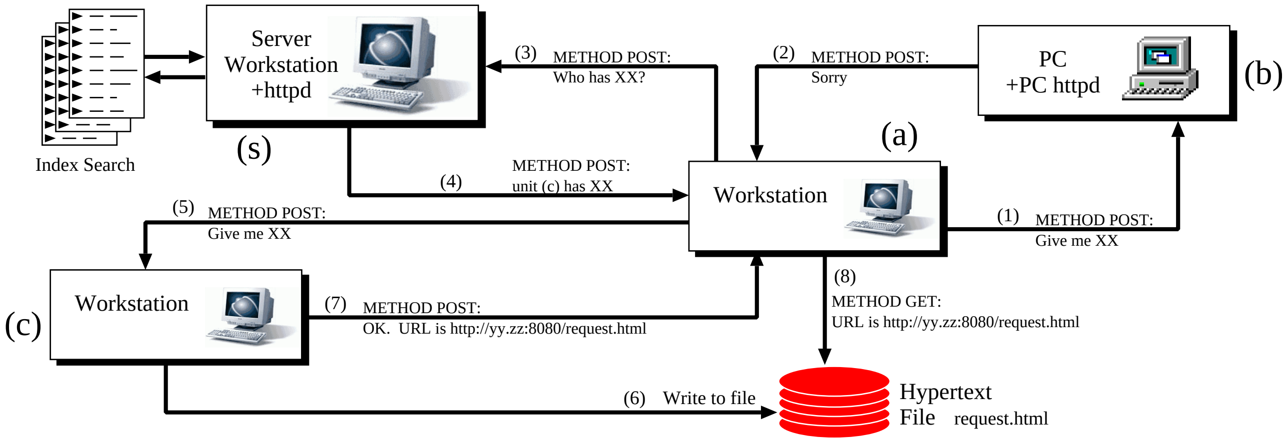

Figure 3: An example of communication in the Ex-W-Pert System.

3.3 Local knowledge-base

The local knowledge-base is composed of a domain-oriented library of the components model and other design documents. A main advantage of the Ex-W-Pert system is that documents and component models related to a particular design task can be focused in a single expert system 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 through *HTTP* communication.

3.4 Communication engine

The communication engine is the heart of the system. It allows the functional reasoning **#QFF** and specification matching **#SME** engines to acquire appropriate knowledge and data from the other sites, if required. Figure 3 shows an example of the communication handled by the communication unit.

All the expert units understand and communicate through *HTTP*. Each unit has a local index of the related information and their site names. In *step 1* a request for additional information from any unit, say unit (a), activates a POST method [9] and asks for the 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 to 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 data 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 directly put the requested URL in the hand of unit (a).

Note that communication through the *HTTP* requires all the transferable information be first written into a hypertext file and then transmitted. Therefore the communication engine is responsible for preparing the information and rewriting the information in the hypertext form before delivering it to the Web system.

In conventional application of the Web 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 designer is not required to be aware of the *HTTP* connections. Therefore, it is not necessary for the user to have networking and Web knowledge in order to use and interact with the system.

4 Discussion

4.1 Other works

Groupware Systems [3] are around for a while, considering computer support of a team work development. There are already examples of groupware applications such as bulletin boards, video conferencing, sharing work processes 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 platforms with various operating systems (DOS, 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.

On the other hand, the idea of implementing and using large scale encyclopedic knowledge bases is struggling for more that two decades without much success. Even if such a large scale knowledge base could be implemented, the cost of data search and retrieval would be tremendously high. We think that the only solution is using a Web-like distributed knowledge base equipped with networking and efficient communication engine. In the above sections we introduced one such architecture by blending the network and expert system technology.

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 these two latter methods is crucial to our project. A disadvantage of *HTTP* is that the above mentioned methods must be launched by using a browser client. In our project, in many cases that two knowledge based systems communicate and exchange information, launching a client software is unwanted and brings unnecessary overhead. We have developed programs to enhance the functionality of the Web methods without launching a client. A platform running *HTTP* daemon can talk to a *WAIS* server. In this project, we have also used the *WAIS* index search capability for efficient searching of the knowledge bases.

5 Conclusion

In this paper, a new application of the World Wide Web technology in artificial intelligence was discussed. An architecture for a Web-based Distributed Expert System (Ex-W-Pert System) was proposed and an 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 in configuring and implementing distributed large scale knowledge bases systems. A generic expert system shell with the communication engine, documentation engine and 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, 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, "Functional Reasoning, Explanation and Analysis: Qualitative Function Formation Technique," *Proc. 6th Annual Conf. JSAI*, Tokyo, Japan, June 1992, pp. 229-232.

[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] K.D. Forbus, "Qualitative Process Theory," *Artificial Intelligence*, vol. 24, pp. 85-168, 1984.

[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] B. Kuipers, "Qualitative Simulation," *Artificial Intelligence*, vol. 29, pp. 289-338, 1986.

[12] `http://www.w3.org/hypertext/WWW/Proxies/Proxies.html`

[13] `http://www.w3.org/hypertext/WWW/Addressing/Addressing.html`

[14] `http://www.w3.org/hypertext/WWW/TheProject.html`