Web Enabled Expert Systems using Hyperlink-based Inference

Yong U. Song[a], Wooju Kim[b], and June S. Hong[c]

a Department of Management Information Systems, Yonsei University
234, Maji, Wonju, Gangwon, 220-710, Korea
Tel: +82-33-760-2340, Fax: +82-33-763-4324, E-mail: yusong@dragon.yonsei.ac.kr

b Department of Industrial System Engineering, Chonbuk National University
664-14, Deokjin, Chonju, Chonbuk, 561-756, Korea

c Division of Business Administration, Kyonggi University
San 94-6 Yiui-Dong, Paldal-Gu, Suwon, Kyonggi, 442-760, Korea

Abstract

With the proliferation of WWW, providing more intelligence to Web sites has become a major concern in e-business industry. In recent days, this trend is more accelerated by prosperity of CRM(Customer Relationship Management) in terms of various aspects such as product recommendation, self after service, etc. To accomplish this goal, many e-companies are eager to embed web enabled rule-based systems, that is, expert systems into their Web sites and several well-known commercial tools are already available in the market. Most of those tools are developed based on CGI so far but CGI based systems inherently suffer over-burden problem when there are too many service demands at the same time due to the nature of CGI. To overcome this limitation of the existing CGI based expert systems, we propose a new form of Web-enabled expert system using hyperlink-based inference mechanism. In terms of burden to Web server, our approach is proven to outperform CGI based approach theoretically and also empirically. For practical purpose, our this approach is implemented in a software system, WeBIS and a graphic rule editing methodology, Expert Diagram is incorporated into the system to facilitates rule generation and maintenance. WeBIS is now successfully operated for financial consulting in the web site of a leading financial consulting company in Korea.

Keywords:

Expert Systems, HTML, Inference, JavaScript, WWW

1. Introduction

With the advent of the customer centric paradigm in business strategy, e-companies go beyond merely making the sale to acquire relative competitiveness in terms of customer satisfaction thereby maximizing the lifetime value of a business relationship [12].

Giving more intelligence to e-commerce sites is popularly recognized as one of effective approaches to increase customer satisfaction in e-commerce sites by allowing the sites to react intelligently or personalize it appropriately to each customer. Especially nowadays, Web enabled rule-based system shells or expert system shells such as Blaze Advisor are taking the major role in making Web sites act more intelligent. This Web enabled rule-based inference technology is applied to various application areas such as intelligent e-mail management, product recommendations, distance learning and training, help desk for technical support, etc.

Most of Web enabled rule-based system shells have been developed using CGI [2] technology but this CGI based systems usually cause relatively high burden to Web server in terms of both required memory and response time. At the worst case, the system may be down and this is very critical problem in most commercial Web sites. One major reason for this is because each individual request to CGI based system requires relatively much higher resources than request for HTML document. Therefore, if we can build a rule-based inference system based on only a set of HTML documents, we are sure it can perform the same inference task much more efficiently than CGI based approach with a given computer resources.

If HTML based approach is better, first upcoming problem is how to build a system to perform the same rule-based inference tasks with only a set of HTML documents. Song and Lee [16] have already shown that a rule can be represented with a set of HTML documents hyperlinked to each other and we can inference the rule by following the hyperlinks. To solve the first raised problem, we proposed a generalized transformation algorithm from a set of rules to a set of hyperlinked HTML documents, which is equivalent to the set of rules in terms of inference.

Second potential issue arising from adopting HTML documents based rule inference is maintenance of rules will be somewhat different from CGI based approach. While CGI based approach can use a classical rule maintenance approach directly to rules, our approach should maintain a set of HTML documents eventually to maintain the rule base. To address this issue, we also propose a framework that can maintain a set of HTML documents systematically. This framework adopts a graphical rule representation mechanism, Expert's Diagram [11], by which a rule base manager can view and edit a set of HTML documents through an easy-to-understand and graphical rule representation model.

Finally, we have designed and implemented our proposed rule transformation algorithm and HTML based rule inference system maintenance framework as an automated system called WeBIS. For the empirical validation of our approach, we also present performance evaluation of our system through comparison study with CGI based approaches.

2. Related Work

We can classify existing methodologies to build Web enabled rule based systems into five categories according to the applied technologies. Table 1 shows those five categories and their summarized technical features in developing Web enabled rule based systems.

Reducing latency time is most critical in success of a Web sites and this can be directly applicable to Web enabled rule-based systems. From this review of related approaches to Web enabled rule-based systems, we have identified reducing latency time that is, reducing burdens caused by rule-based systems to Web server is still most important but not

Table 1 – Categories of Web enabled rule based systems

Location of inference engine	Type of inference engine	Technical features
Server side	CGI program	• Uses CGI (Common Gateway Interface) standard. • Web server invokes a CGI program while passing required parameters according to CGI standard.
	Server-side script	• Inference engines are developed under environments such as JSP, ASP, and PHP. • Transaction processing and multi-threading functions are provided by default.
	Web server embedded module	• Inference engines are embedded into Web server as a sub-module using API such as NSAPI.
Client side	External viewer	• Is developed as an independent program. • Is invoked by Web browser according to predefined MIME type.
	Java applet	• Bytecodes for inference engine is located in the server side but is transmitted to a Web browser. • JVM in the client side interprets the bytecodes and executes it.

fully resolved issue. This means it is still meaningful to keep making an effort to improve efficiency of Web enabled rule-based systems in terms of latency time.

Under the premise that we can make a rule-based system only with hyperlinked HTML pages and there are no functional differences between HTML based system and conventional Web enabled rule-based systems approaches, HTML based system is definitely much more efficient than the conventional approaches using CGI program or server-side scripts in terms of latency time. But if we want to take this advantage of HTML based system, we have to satisfy the premise first. To satisfy the premise, we need to show how a rule-based system can be implemented with a set of hyperlinked HTML pages. In addition to this, we have to secure a general-purpose mechanism to transform a set of rules to a set of functionally equivalent hyperlinked HTML pages for practical application of this approach to real world cases.

If we can address both prerequisite conditions above, we can take full advantage of HTML based system in Web enabled rule-based inference task. The following section discusses our proposed approach to satisfy both prerequisites.

3. Hyperlink-Based Inference Systems

3.1 Hyperlink-Based Inference

If we want to use a hyperlink-based inference system instead of a rule-based system, first we have to prove the hyperlink-based inference system is functionally equivalent to conventional rule-based systems with any given rule-base. In this paper, definition of rule-base that we would deal with is restricted only to backward style rule. That is, set of rules are devised to deduce eventually one or a group of goals. Now, let's assume we have the following two rules to prove a proposition *c*:

Rule #1: If *a* And *b* Then *c*

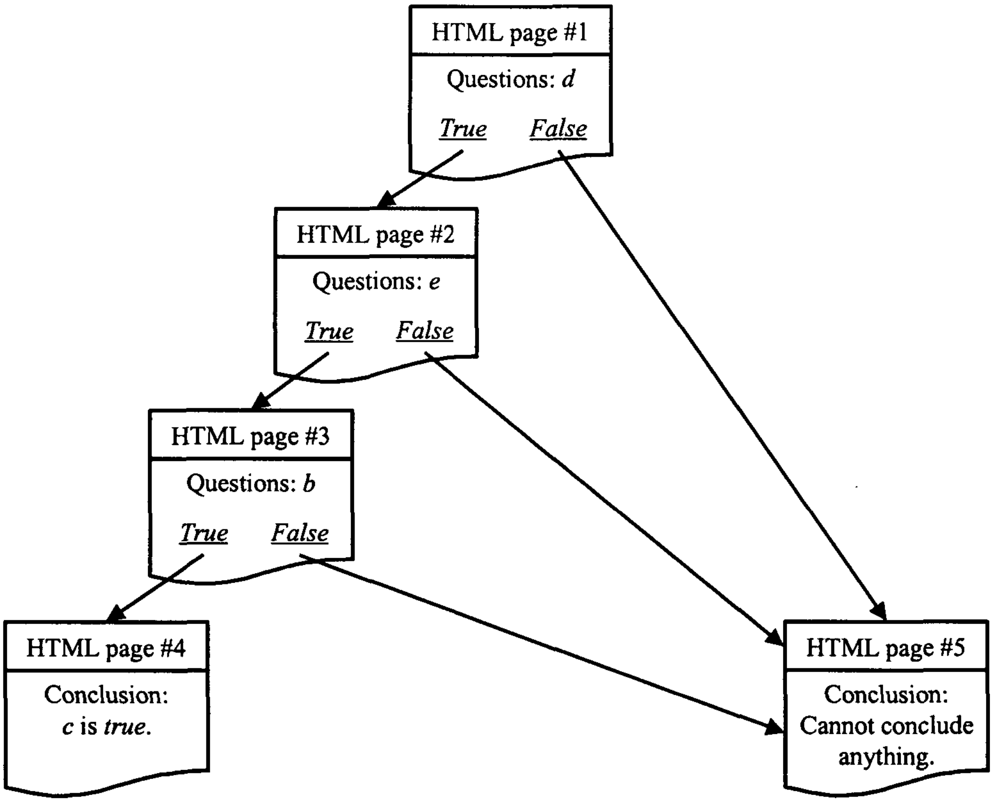


Figure 1 – An Example of Functionally Equivalent Set of Hyperlinked HTML Pages

Rule #2: If *d* **And** *e* **Then** *a*

If we apply these two rules to a typical rule-based system with the mission to prove the proposition *c*, it will ask a user about *d*, *e*, and *b* consecutively under the assumption that applied backtracking method follows a left-to-right rule. If the result from all these three questions is that they are all *true*, then *c* is proved to be *true*. Otherwise, if any one of the answers is *false*, the rule-based system cannot conclude anything in this case. Related to two above rules, we can emulate the same series of questions and conclusions as the typical rule-base systems did, by using a set of hyperlinked HTML pages. The set of hyperlinked HTML pages for our example is illustrated in Figure 1 and the readers can easily identify that these hyperlinked HTML pages can do exactly the same inference task as the rule-based systems. Let's define such a functionally equivalent set of hyperlinked HTML pages to a given set of rules in terms of inference as *Functionally Equivalent hyperlinked HTML page Set* (*FES*) and it is denoted by *FES*(*x*) for a given set of rules *x*.

In the case of the variable having numeric type domain, it is not so simple to construct a corresponding *FES*() because we have to obtain the exact numerical value first from a user and obtaining a numerical value is practically not possible by hyperlinks between HTML pages. Of course, in the case that there is only a set of finite number of ranges for value exclusive to each other, we may construct *FES*() in a similar way to the case of symbolic type domain by regarding each range as a separate hyperlink. But in a general sense, since we may need to compute a function based on the value of the variable and to compare it with a constant or the other

```
<HTML>
<HEAD>
<TITLE> Expression Type </TITLE>
<SCRIPT LANGUAGE="JavaScript">
        <!--
          function verifyValue(form)
          {
                    if (form.total_income.value >= 0.2 * form.threshold.value)
                              location="tax.html"
                    else
                              location="notax.html"
          }
        //-->
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="exprform">
Input the values of the following variables... <BR>
total income <INPUT TYPE="text" NAME="total_income"> <BR>
threshold <INPUT TYPE="text" NAME="threshold"> <BR>
<INPUT TYPE="button" VALUE="OK" onClick="verifyValue(this.form)">
</FORM>
</BODY>
</HTML>
```

Figure 2 – Source of JavaScript Embedded HTML page for Numeric Type Variables

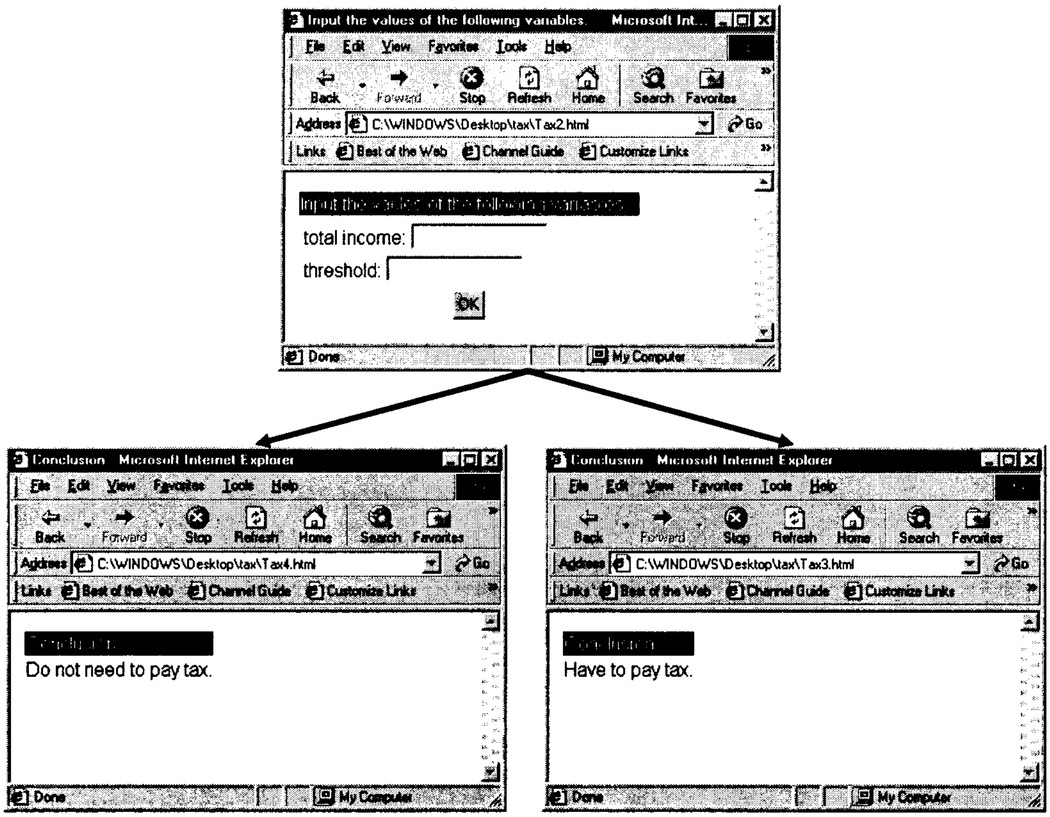
variables, we can say only hyperlinks is not enough to represent the rules containing numeric type of variables.

But one simple way to cope this limitation of hyperlinks is to use a client-side script language such as JavaScript [1] and VBScript [10] in obtaining the values of the variables and computing and comparing them if needed. Suppose there are the following two rules related to a tax consulting.

Rule #3: **If** *total income* ≥ 0.2 * *threshold*
Then Have to pay tax.
Rule #4: **If** *total income* < 0.2 * *threshold*
Then Do not need to pay tax.

From these two rules, we can identify two numeric type variables *total income* and *threshold* and we also find numerical computations and comparisons must be performed to inference. Using JavaScript, we can construct *FES*({Rule #3, Rule #4}) with three hyperlinked HTML pages; one JavaScript embedded page for the premise part and two simple HTML pages for the conclusions. Figure 2 illustrates the required source of the first JavaScript embedded page. This source is then divided into two parts, form part and script part as shown in the figure. The form part is for obtaining the values of the numerical variables in rules. The script part performs computations and comparisons of the obtained values required in the premise of the rule and then branches to the corresponding page for conclusion depending on the results from the comparisons. Especially the script part is written in bold face to distinguish it from the form part in the figure 2. Figure 3 also illustrates example screen shots generated by *FES*({Rule #3, Rule #4}).

Of course, which screen will be activated between two appeared in the lower side in Figure 3 is depending on the computed result performed in the upper part screen in the figure based on the user's


Input the values of the following variables - Microsoft Int...
C:\WINDOWS\Desktop\tax\Tax2.html
Conclusion - Microsoft Internet Explorer
C:\WINDOWS\Desktop\tax\Tax4.html
C:\WINDOWS\Desktop\tax\Tax3.html
Back
Forward
Stop
Refresh
Home
Search
Favorites
Links
Best of the Web
Channel Guide
Customize Links
Done
My Computer

Input the values of the following variables

total income:

threshold:

OK

Conclusion

Do not need to pay tax.

Conclusion

Have to pay tax.

Figure 3 – Example Screen Shots Generated from FES({Rule #3, Rule #4})

inputs. So far, we have shown with several examples that typical but various type of rules can be transformed to the functionally equivalent hyperlinked pages, that is, *FES*()s. But in spite of such examples, we are not sure if we can find *FES*() functionally equivalent to any arbitrary set of rules, yet. To make sure that there exist *FES*() to any arbitrary set of rules and also we can find it, we propose a *FES* generation algorithm for an arbitrary set of rules. This algorithm is addressed in the next subsection subsequently.

3.2 FES Generation Algorithm

Basically our FES generation algorithm takes an arbitrary set of rules and transforms it to a functionally equivalent set of hyperlinked pages. To achieve this, FES generation algorithm consists of two major phases. In the first phase, we convert each rule of an arbitrary rule set to the corresponding canonical form, called SNF (simplified normal form). Based on the converted rules in the canonical form, the second phase generates functionally equivalent set of hyperlinked pages (*FES*) in terms of inference.

As a simple example, let the following two rules in SNF be a rule set necessary to be transformed to *FES*.

$$R_1: A \wedge B \wedge C \Rightarrow X$$

$$R_2: B \wedge D \Rightarrow Y$$

And, figure 4 illustrates how FES generation proceeds to generate hyperlinked web pages with example rules at step by step level.

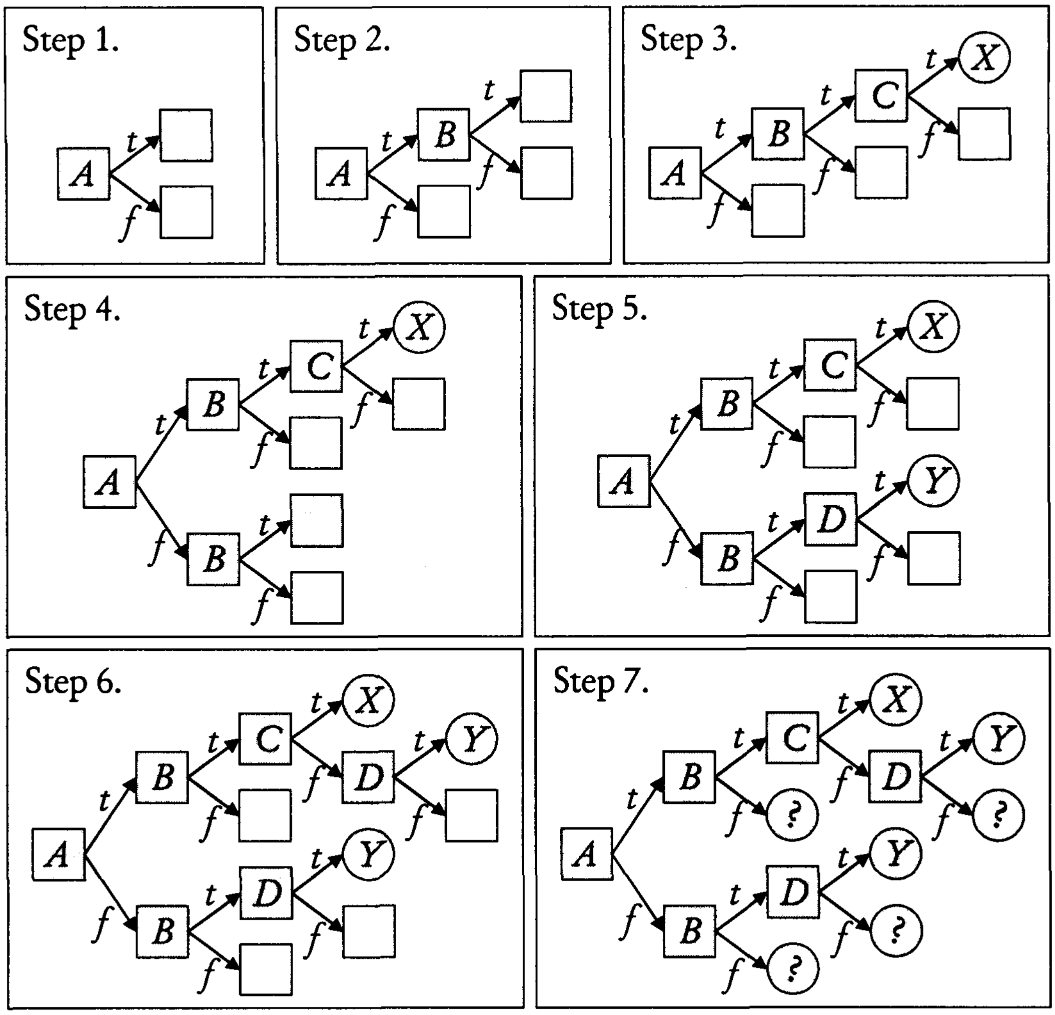


Figure 4 – Progress of FES Generation Algorithm with the Example Rules

3.3 Algorithm Extension

The algorithm proposed in the previous section can cover only so called 'Fact Type' rule mentioned earlier. There are two additional popular rule types, rules with symbolic variables and numerical variables. But with minor revision on the FES generation algorithm, we can easily take into account those types of rules. We are not going into the details of revision here since it is so trivial.

4. WeBIS Approach

So far, we have proposed the idea that hyperlinked web pages can perform inference tasks and have shown how a rule base can be automatically converted to functionally equivalent set of hyperlinked web pages (FES). To facilitate our approach, we provide system architecture, Web Based Inference System (WeBIS), which can support users to develop a set of hyperlinked web pages to perform an inference task systematically. Figure 5 shows the overall architecture of WeBIS and its three

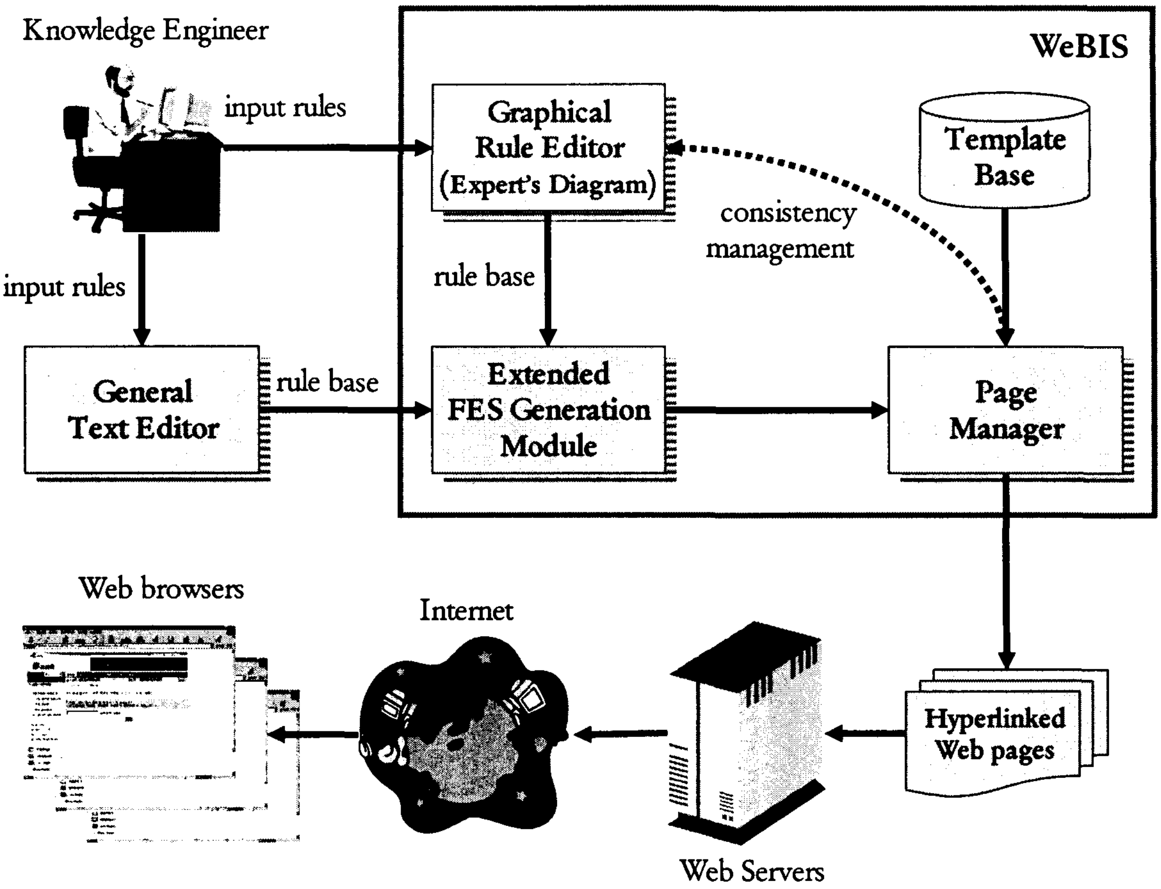


Figure 5 – Architecture of WeBIS

major components. Major information flows are also depicted. As shown in Figure 5, knowledge engineer can input rules to the system via both general text editor and graphical rule editor which is provided by WeBIS. Then, WeBIS transforms rules to functionally equivalent set of hyperlinked web pages. Those pages are registered to Web servers and finally users can access them and get inference services via Internet.

5. Performance Evaluation

To prove the efficiency of our approach, we compare the response time of hyperlink-based inference system with those of inference systems which were developed by using conventional commercial RBMS (rule base management systems). To evaluate the performance of each system, we count the total number of responses per 3 minutes from Web server when 10 client computers send requests simultaneously via Web. The hardware specification of the server computer is Pentium 4 CPU of 1.5GHz with 256MB of main memory. The OS of the server is Windows 2000 Server. The hardware specification of all of the client computers is Celeron CPU of 633 MHz with 238MB of main memory. The OS of the client is Windows 98. Because the commercial RBMS deploy their application systems using JSP (Java Server Pages), we use Tomcat as the Web server. Table 2 shows the result of the performance evaluation.

The total number of responses of our approach is 75,176 per 3 minutes while those of two commercial tools are 335 and 9,181 respectively, and the ratios are 224.54 and 9.18. Hence, we can conclude that our

Table 2 – Performance Evaluation Result

Unit: #responses per 3 min.

Client	WeBIS	A*	Ratio	B*	Ratio
PC1	7,471	34	219.74	932	8.02
PC2	7,568	34	222.59	931	8.13
PC3	7,496	33	227.15	900	8.33
PC4	7,563	33	229.17	921	8.21
PC5	7,534	32	235.44	895	8.42
PC6	7,435	35	212.43	893	8.33
PC7	7,486	34	220.16	913	8.20
PC8	7,189	33	217.83	930	7.73
PC9	7,681	34	225.90	947	8.11
PC10	7,755	33	235.00	919	8.44
Sum	75,176	335	224.41	9,181	8.19
Average	7,518	34	224.54	918	8.19

* Conventional RBMS A and B

approach is about 9 ~ 225 times faster than conventional approaches.

6. Conclusions

With the proliferation of WWW, providing more intelligence to Web sites has become a major concern in e-business industry. In recent days, this trend is more accelerated by prosperity of CRM(Customer Relationship Management) in terms of various aspects such as product recommendation, self after service, etc. To accomplish this goal, many e-companies are eager to embed web enabled rule-based systems, that is, expert systems into their Web sites and several well-known commercial tools are already available in the market. Most of those tools are developed based on CGI so far but CGI based systems inherently suffer over-burden problem when there are too many service demands at the same time due to the nature of CGI. To overcome this limitation of the existing CGI based expert systems, we propose a new form of Web-enabled expert system using hyperlink-based inference mechanism. In terms of burden to Web server, our approach is proven to outperform CGI based approach theoretically and also empirically. For practical purpose, our this approach is implemented in a software system, WeBIS and a graphic rule editing methodology, Expert Diagram is incorporated into the system to facilitates rule generation and maintenance. WeBIS is now successfully operated for financial consulting in the web site of a leading financial consulting company in Korea.

References

[1] Danesh, A., *Teach Yourself JavaScript in a Week* Sams.net Publishing, 1996.

[2] Dwight, Jeffry, Michael Erwin, and Robert Niles, *Special Edition Using CGI*, Second Edition, QUE, 1997.

[3] EXSYS, Inc., *Moving an EXSYS Application to the EXSYS Web Runtime Engine (WREN).*

[4] Fielding, R., J. Gettys, J. C. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, *Hypertext Transfer Protocol -- HTTP/1.1*, INTERNET-DRAFT <draft-ietf-http-v11-spec-rev-05>, Internet Engineering Task Force, 11 September 1998.

[5] Giarratano, Joseph and Gary Riley, *Expert*

Systems: Principles and Programming, 2nd Edition, PWS Publishing Company, 1994.

[6] Jerke, N., Hatmaker, M., and Anderson, J., *VBScript Interactive Course*, Waite Group Press, 1997.

[7] Lee, J. K., I. K. Lee, and H. R. Choi, "Automatic rule generation by the transformation of Expert's Diagram: LIFT," *Int. J. Man-Machine Studies*, 32(1990), 275-292.

[8] McCarthy, J. C., "The Social Impact of Electronic Commerce," *IEEE Communications Magazine*, vol. 37, no. 9, 1999, pp. 53-57.

[9] Nilsson, N. J., *Principles of Artificial Intelligence*, Springer-Verlag, 1982.

[10] Raggett, Dave, Arnaud Le Hors, and Ian Jacobs(ed.), *HTML 4.0 Specification REC-html40-19980424*, W3C, 24 April 1998.

[11] Russell, S. and P. Norvig, *Artificial Intelligence – A Modern Approach*, Prentice Hall, 1995.

[12] Song, Y. U. and Lee, J. K., "Automatic Generation of Web-based Expert Systems," *Journal of Intelligent Information Systems* (in Korean), vol. 6, no. 1, 2000, pp. 1-16.

[13] http://www.blazesoft.com/

[14] http://www.expertise2go.com/

[15] http://www.exsys.com/

[16] http://www.ilog.com/