Development of an Intelligent Management System for Monitoring Educational Web Servers

Gwo-Jen Hwang
Information Management Department
National Chi Nan University
Pu-Li, Nan-Tou,
Taiwan 545, R. O. C.
FAX: 886-49-2915205
TEL: 886-915396558
E-mail: gjhwang@ncnu.edu.tw

Judy C.R. Tseng, Chih-Hsiang Wu & Chung-Ming Li
Department of Computer Science and Information Engineering
Chung Hua University
Hsinchu, 300, Taiwan, R.O.C.
TEL: 886-915396565
E-mail: judycrt@chu.edu.tw

Gwo-Hour Hwang
Information Management Department
Ling Tung College
Taichung, Taiwan 400, R.O.C.
TEL: 886-915392150
ghhwang@mail.ltc.edu.tw

Abstract
In recent years, distance education has become an important issue that attracts the researchers from both computer science and education fields. Although many systems were proposed to support distance learning activities, few of them took into consideration web server management issues that are critical to the stability and reliability for providing educational service. To cope with these problems, this study proposes a novel approach to eliciting and integrating system management knowledge from experienced system managers. An intelligent management system for monitoring educational web servers has been implemented based on the novel approach. The system is capable of predicting and handling possible failures of educational web servers, and hence the stability and reliability of the web servers can be significantly improved.

Keywords: Internet service, Distance education, System management, Digital learning, Web servers

1. Introduction
The advent of information technology has led to the development of new educational styles and many computer-based educational systems have been developed in recent years. In 1989, Johnson et al. presented MITT (Microcomputer Intelligence for Technical Training) Writer, an authoring environment for building intelligent tutoring systems for computer courses, which represented a practical application of Artificial Intelligence (AI) in technical training (Johnson et al. 1989). Later, Gonzalez and Ingraham (1994) designed an intelligent tutoring system, which is capable of automatically determining assignment progression and remediation during a training session, according to past student performance. Vasandani and Govindaraj (1995) developed an intelligent tutoring system that assists in the organization of system knowledge and operational information in order to enhance operator performance.
Meanwhile, Harp et. al. (1995) employed the technique of neural networks to model the behaviors of students, in the context of intelligent tutoring systems, using self-organizing feature maps to capture the possible states of student knowledge from an already existing test database. Furthermore, Rowe and Galvin (1998) employed planning methods, consistency enforcement, objects and structured menu tools to construct intelligent simulation-based tutors for procedural skills. Antao et al. (2000) presented their experiences in relation to the developing intelligent tutorial systems for teaching simulation in engineering education.

While computer intelligence has inspired the development of intelligent tutoring systems however, the advent of network technologies in the mean time has helped the further realization of distance learning systems. For example, Sun and Chou (1996) presented the CORAL (Cooperative Remotely Accessible Learning) system, which is aimed to institute a collaborative learning environment on computer networks. One of the branches of CORAL is the Intelligent Tutoring and Evaluation System (ITES) project, which focuses on applying the techniques of AI to enhance the tutoring process (Hwang 1998).

Although many distance educational systems have been developed, few have taken into consideration the issues of stability and reliability. To manage these problems, an intelligent management system has been implemented to predict and resolve any possible failures of educational web servers. A knowledge engineering approach was employed to elicit the knowledge of system management from domain experts. Experimental results have indicated that the stability and reliability of the web servers could be significantly improved with the aid of the appropriate management system.

2. Literature Review
Information technology has been acknowledged as an important component in making management-level plans. Knowledge engineering is one of the most popular information technologies that have been applied to a wide variety of applications, such as the development of decision-making systems, monitoring systems, expert systems and intelligent tutoring systems (Orr et al. 1997, Hahn and Subramani 2000).

Knowledge elicitation is known to be a critical bottleneck in the building of knowledge-based systems, and the Repertory Grid test is one of the most important and popular methods when coping with the knowledge elicitation problems (Kelly 1995). A Repertory Grid is a two way classification of data in which events are interlaced with abstraction in such a way as to express part of a personal system of cross-referencing between observations or experience of the world and the constructs or classifications of said experience (Shaw and Gaines, 1987; Janice and Robert, 1991). In 1995, a revised method of Repertory Grid test, the Fuzzy Table approach, was proposed (Hwang 1995). The process of constructing a Fuzzy Table is similar to that of a Repertory Grid. The major differences are the use of fuzzy variables instead of constructs, and a pair of values, i.e., a rating to indicate the most desirable fuzzy value and the degree of certainty for giving that rating, for each entry instead of a rating.

In building a reliable knowledge-based system, usually the cooperation of multiple experts is required. Several algorithms have been proposed to integrate knowledge elicited from multiple experts, such as MERGE (Hwang 1994), fuzzy table integration (Hwang 2002) and GA-based knowledge-integration framework (Wang et al. 2000).

3. Elicitation of System Management Knowledge
System management is a very important issue to ensure the stability and reliability of a computer system, especially for a web server. A system manager’s knowledge is primarily
gathered from past experiences of handling system failures. As the possible reasons of system failures for a web server may relate to the status of computer networks, hardware, software, users and system operators, it is difficult to precisely predict any possible failures and prevent such failures before they happen, even an experienced system manager can monitor the performance of an entire system twenty-four hours a day and still be unable to prevent system failure. Therefore, it is a challenging issue to develop an intelligent system that monitors the performance of the web server whilst managing any possible failures in advance. We shall present the process of eliciting the knowledge of identifying possible failures of educational servers by applying a hybrid knowledge elicitation procedure that combines the notations described in the previous section (Pan 2002):

(1) Define the Objects of Identifying Possible Failures
An object may denote the problem to be detected, the goal to be achieved, the decision to be made, the entity to be classified, or the action to be performed. Considering the problem domain of ‘Possible failures of Educational Servers’, the objects can be the possible problems to be detected, such as ‘system halt’, ‘network failure’ and ‘hardware malfunction’.

(2) Elicit the Prosperities for Describing the Objects
A concept grid is a collection of concepts that can be used to describe the priorities of the object. For the problem domain of ‘Possible failures of Educational Servers’, there are several relevant factors concerning server failures, such as ‘computer virus’, ‘disk space’ and ‘memory space’. Experts with different backgrounds and experiences usually use their own familiar factors to define possible system failures. For example, if we want to elicit and collect knowledge from domain experts to predict if system failures will happen, an illustrative example given in Table 1 depicts a concept grid provided by an expert.

Table 1: Illustrative Example of a Concept Grid

<table>
<thead>
<tr>
<th>Concept Name</th>
<th>Unit</th>
<th>Linguistic Form</th>
<th>Membership Function</th>
<th>Concept Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD space</td>
<td>M Bytes</td>
<td>Sufficient/Normal/Insufficient</td>
<td>(50, 30, 10)</td>
<td>Free disk space</td>
</tr>
<tr>
<td>Memory space</td>
<td>M Bytes</td>
<td>Sufficient/Normal/Insufficient</td>
<td>(500, 300, 100)</td>
<td>Free memory space</td>
</tr>
<tr>
<td>Computer virus</td>
<td>Rank of danger</td>
<td>1 / 2 / 3 / 4 / 5</td>
<td>(1, 3, 5)</td>
<td>Virus Information of Trend Micro</td>
</tr>
<tr>
<td>H/W Damage</td>
<td>Boolean</td>
<td>True / False</td>
<td>N/A</td>
<td>Status of computer hardware</td>
</tr>
<tr>
<td>Power failure</td>
<td>%</td>
<td>High / Middle / Low</td>
<td>(10%, 30%, 50%)</td>
<td>UPS power remain</td>
</tr>
<tr>
<td>Programme Bug</td>
<td>Boolean</td>
<td>True / False</td>
<td>N/A</td>
<td>Operative condition of the programme</td>
</tr>
<tr>
<td>OS Failure</td>
<td>Boolean</td>
<td>True / False</td>
<td>N/A</td>
<td>The server operative condition</td>
</tr>
<tr>
<td>CPU loading</td>
<td>%</td>
<td>High / Middle / Low</td>
<td>(100%, 90%, 80%)</td>
<td>CPU loading</td>
</tr>
</tbody>
</table>

(3) Obtain the Integrated Concept-Object Relationships
As two experts may use different concept names to represent the same concept, it is necessary to negotiate the experts to derive a common concept grid structure by asking them to check whether the vocabularies and values in those concept grids are consistent.

(4) Calculating the Weight of Each Expert
As each expert may have different experiences and backgrounds, it is necessary to define the weight of each expert based on relevant criteria, such as ‘past experiences’, ‘background relevance’ and ‘relevant credits’.

(5) Calculating the Relative Importance of Each Concept
In this phase, the fuzzy pair-wise comparison matrix method is employed to assist experts to determine the relative degree of importance for each concept (Satty, 1980; Laarhoven and Pedrycz, 1983).

(6) Rating the Concept-Object Grid
After deriving the common concept grid from multiple experts, the knowledge integration module forms a ‘Concept-Object Grid’ in which the ‘Concept Name’ column is identical to the common concept grid.

(7) Integrate Fuzzy Knowledge and Generate Rules
Based on the integrated knowledge, a set of fuzzy rules is generated. For example:

IF HD space is not enough AND Memory is not enough AND Computer virus is very dangerous AND H/W damage is more or less serious AND Power failure is very serious AND Programme bugs exists AND OS failure is very serious AND Network loading is more or less Low AND CPU loading is very low AND Hacker intrusion warning is more or less high

THEN System status = ‘Server crash’ with CF = 0.7833

4. Implementation and Evaluation
The management system was developed on a server equipped with P4 2.4 G CPU, 512 MB DDR 333 RAM and 80GB hard disk. Figure 1 shows the homepage of the educational server, ITED (Intelligent Testing, Evaluation and Diagnosis) system, which is capable of diagnosing student learning problems and providing personalized learning suggestions.

Figure 1: Interface of ITED system

Figure 2 shows the control panel of the management system, which is used to specify the educational servers to be monitored.
Figure 2: Control panel of management system

Figure 3 shows an illustrative example of applying the management system to monitor the CPU status for an educational website. Figure 4 shows the system records of detecting and resolving possible system failures.

Figure 3: Illustrative example of applying the management system to monitor the CPU status of an educational website

To evaluate the performance of the management system, a one-week experiment was conducted. A programme was employed to occupy the resources of the educational servers to produce system failures. For example, it will increase the loads of CPU and network to 100% for three minutes, and will repeatedly create and remove files to reduce continuous free disk space. Two copies of the educational server were used in the experiment: the supervised server was monitored by the management system and the unsupervised server was not. Table 2 depicts the experimental results.
Figure 4: System records of detecting and resolving possible system failures

Table 2: Experimental results

<table>
<thead>
<tr>
<th></th>
<th>Supervised server</th>
<th>Unsupervised server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Times of server failures</td>
<td>23</td>
<td>2549</td>
</tr>
<tr>
<td>Total time of system malfunction</td>
<td>31 minutes</td>
<td>3447 minutes</td>
</tr>
<tr>
<td>Number of system failures predicted and resolved by the management system</td>
<td>1625</td>
<td>0</td>
</tr>
<tr>
<td>Total time needed for resolving malfunctions manually</td>
<td>Approx. 12 hours</td>
<td>Approx. 78 hours</td>
</tr>
</tbody>
</table>

From the record of ‘times of server failures’ and ‘Total time of system malfunction’, it can be seen that the stability of the supervised server was significantly superior when compared to the unsupervised one. The stability improvement is due to the automatically malfunction-resolving feature, which has predicted and resolved 1625 possible system failures before they actually occurred. Furthermore, the total time needed for resolving malfunctions manually was significantly reduced (over six times, from 78 hours to 12 hours), which showed that the intelligent management system could substantially ease the load of the system manager. To sum up, we conclude that the novel approach is desirable.

5. Conclusions

In this paper, an intelligent management system is proposed to monitor the performance of educational web servers. The management system is capable of predicting and resolving potential problems of the educational web servers, enabling the stability and reliability of the servers to be improved. Some experiments, applying the novel approach to monitor educational web servers, have achieved promising results. It can be seen that the novel approach could also be applied to the management of other computer servers, such as database servers, e-business servers, mail servers, etc. As a result, we are planning to expand the knowledge base, so that the management system can be applied to various types of servers allowing more potential problems to be successfully managed.
Acknowledgement

This study is supported in part by the National Science Council of the Republic of China under contract numbers NSC-92-2520-S-260-001 and NSC-92-2524-S-260-002.

References


