International Journal of Computer Science & Applications
Vol. 5, No. 2, pp 89 - 104

THE DESIGN OF DECOMPOSED SCORM STRUCTURE WITH EMBEDDED LCMS BROKER

Fu-Chien Kao

Department of CSIE, Da-Yeh University,NO.112, Shanjiao Rd., Dacun, Changhua,51591,Taiwan,R.O.C.
fuchien@mail.dyu.edu.tw

Chih-Wei Tseng, Wen-Yu Chang, Chang-Yu Huang

Computer Center,Institute of CSIE, Da-Yeh University,NO.112, Shanjiao Rd., Dacun, Changhua,51591,Taiwan, R.O.C.
{Reason,R9606009, R9506001}@mail.dyu.edu.tw

This study proposed a decomposed SCORM (Sharable Course Object Reference Model) structure with embedded LCMS Broker that provides the integration of learning resources and balance of network traffic. The system consists of the Learning Management System (LMS) for processing basic data, learning data and learning records of learners, and the Learning Content Management Systems (LCMSs) for managing and storing course resources. The Web Service cross-platform distribution configuration of this study provides common communications between systems and enhances the capability of integrating learning resources. The proposed embedded LCMS Broker ensures a load-balancing functionality for LCMS of different domains. The connection program embedded into the PC of the learner via LMS connects to the embedded LCMS Broker for access to required teaching materials from each LCMS. The LCMS with the minimum load is then selected from suitable LCMS as the source of the teaching materials. Since the SCORM standard integrates teaching materials and platforms using JAVA Script, cross-domain scripting issues may occur when the learning system and course resources are stored in subsystems of two different domains. This study proposed a SCORM learning environment by creating a cross-domain server with URL-rewrite technology to provide a solution for this issue.

Keywords: Embedded Broker, SCORM, Load-Balancing, LCMS

1. Introduction

With the popularization of the Internet, both the academic community and industry increasingly utilize e-Learning and related technologies. To integrate learning resources of different platforms and share teaching materials easily using these platforms, designers of teaching materials and developers of platforms must observe the same standards from the earliest stage of the design and development. Hence, the SCORM standard is created and all teaching materials conforming to this standard can be easily transferred to any platforms developed using the same standard. Similarly, different SCORM teaching materials can be imported from platforms based on the same standard. The reusability, accessibility, durability, interoperability, adaptability, affordability, and manageability of teaching materials are thus ensured and users may access more resources under this structure (Hsu et al., 2003).

Standardization of format solves the problem of teaching material transferral, but courses cannot be shared amongst individual learning management systems. Most current research focuses upon the development of functions and teaching materials based on a single learning platform (Tresso; Liu et al.; Tung, 2002), rather than the integration of multiple platforms for users to access course resources from a single portal. Even though there is a good dependency between teaching materials and platforms, the user who needs to transfer resources between platforms must import data from individual courses in each platform. Sharing efficiency in the circumstances is, thus, significantly affected. To solve this problem, a common standard is required for the integration of different platforms and teaching materials to ensure sharing of resources and optimization of their utilization.

Besides, the course transfer between platforms should be implemented in a real-time and efficient manner. However, this is inefficient if the importation of teaching materials and transfer of files are executed when the learner cannot find required teaching resources on the platform in use and identifies the resources on another platform via the common interface. It takes even more time for the transfer of huge teaching material files when audio/video multi-media are used. In addition to the common interface between platforms, the architecture of the plant must also be improved.

The proposed system based on a decomposition structure consists of two sub-systems (Kao et al., 2006; Chien et al., 2005): The Learning Management System (LMS) for processing basic data, learning data and learning records of students, and the Learning Content Management System (LCMS) for managing and storing course resources. The two sub-systems can be built on different servers and provide common communications between systems via the Web Service cross-platform distributing configuration. When logging in the LMS, users can explore required teaching materials from distributed LCMS and select the most suitable LCMS based on the decision of the proposed embedded LCMS Broker. Providing service directly from the LCMS to the client solves the problem of transferring teaching material files between platforms.

In this paper, we propose a SCORM structure that effectively shares LCMS network traffic by the load-balancing function of embedded LCMS Broker. The structure is capable of creating an integrated SCORM learning environment by the collaboration of two servers in different domains to improve the resource integration capability of the SCORM and the response efficiency of the system.

2. Standard and Protocol

This study builds a distributing learning system in conformity to the SCORM standard under the Web Service structure. Related standards and protocols are SCORM and Web Service.

2.1. *SCORM Standard*

To shorten the time-to-market and cost of teaching materials and ensure their reuse and circulation for different LMS by building the “Teaching Material Reuse and Share

Mechanism", the Whitehouse Technology Office and Department of Defense (DoD) of the United States initialized the ADL (Advanced Distributed Learning Initiative) in late 1997. Teaching material developers, users, and institutions responsible for proliferating IMS2, AICC (Aviation Industry CBT Committee) and IEEE standards participated in this program to develop an associated set of technology guidelines – SCORM based on existing achievements in the creation of relevant standards (Wang et al., 2002). The standard mainly defines two system structures: "Content Aggregation Model" (CAM) and "Run-time Environment" (RTE). These structures are briefly described as follows (Dodds et al., 2001).

2.1.1. *Content Aggregation Model (CAM)*

In past e-Learning environments, the transmission model between teaching materials and platforms differs depending on the structure of individual platforms. As a result, teaching materials can only be executed on platforms developed by the same manufacturer and copying original data to other platforms is costly. To solve the problem of teaching materials that cannot be reused by platforms due to different data formats, SCORM provided the Content Aggregation Model as a standard for the development of teaching materials (Wang et al., 2002).

CAM defines a picture or music file in the teaching material as an Asset. A collection of Assets is referred to as a SCO (Sharable Content Object), such as a web page containing texts, pictures and music files. Metadata files describe the component (Asset, SCO) of the teaching material in XML format for further management of course resources. Content Packaging uses Manifest files, with the same file name imsmanifest.xml, for packing teaching materials and describes members of the teaching material and course structures in XML format(Liu et al.; Shih et al., 2003). As shown in Fig. 1. it shows a structure that conforms to the SCORM standard. Once a course is exported as Content Package, the LMS that conforms to SCORM is able to analyze the Manifest file (imsmanifest.xml) of the Content Packaging and import the course in the platform for sharing (Chu et al., 2004; Wang et al., 2002; Dodds et al., 2001).

2.1.2. *Content Packaging and Meta-Data*

Content Packaging is used to provide description and packaging material (Dodds et al., 2001). Content Packaging includes list file and main file parts. The list file uses XML language to describe the contents, structure, location, and content type for the main file. Based on the list file defined structure sequence and assigned course files, the learning management system can enable the user to browse and request courses (Yang et al., 2003) from the packaged compressed file (.ZIP).

Meta-Data mainly records the summary of contents, providing the LMS criteria to check whether material for the user is correct (Dodds et al., 2001). Component contents from Assets, SCO, to individual courses all has to be summarized by Meta-Data to provide the LMS to search through and reuse the teaching material referencing sources (Dodds et al., 2001).

2.1.3. *Run-Time Environment (RTE)*

Most existing e-Learning systems are developed with different languages and the definition of delivered parameters is different, indicating a lack of common communication between platforms and teaching materials. Run-Time Environment uses Java Script for the integration of platforms and teaching materials and creates communications between them via APIWrapper.js and SCOFunctions.js; the former records the current status of courses, while the latter records the performance of each learner. The platform will store this data in the database and the integration of platforms and teaching materials is implemented using the same applications. Fig. 1. is the schematic drawing of the Run-Time Environment. The user connects to the LMS from his or her PC and embeds the API Adapter downloaded from the LMS in the browser of the PC. The API Adapter is driven by two Java Script applications contained in each course. It is a medium for communication between the LMS and teaching materials and responsible for the transmission and reception of data between the client and server (Wang et al., 2002).

2.1.4. *Teaching Material Sequence Guide*

Teaching material sequence guide mainly uses the XML language to describe the materials. The goal of the guide (ADLNet, 2005) is to link the individual learning objects that conform to SCORM 2004 SN, to be able to adapt to differences between different users. The teaching material and objects can then be mixed and matched to conform to users learning conditions, teaching students in accordance with their aptitude. Supported teaching material sequence guide for SCORM includes eight syntaxes, Control Mode, Sequencing Rules, Limit Conditions, Auxiliary Resources, Rollup Rules, Objectives, Randomization Controls, Delivery Controls etc. (ADLNet, 2005).

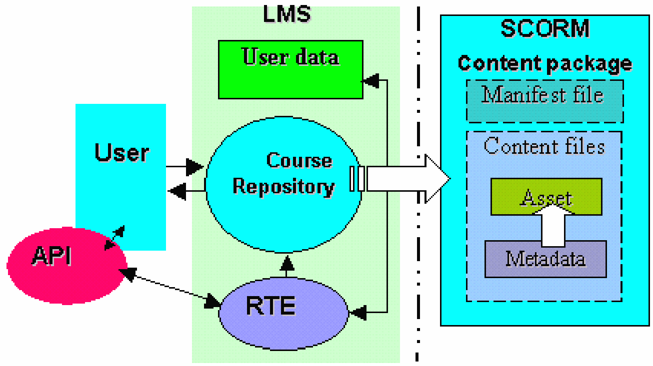

Fig. 1. SCORM Structure.

2.2. *Web Service Structure*

Web Service is an application component comprised of SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language), and UDDI (Universal Description Discovery Integration), and can be accessed via the Internet based on XML, SOAP, or HTTP format (Liu et al.; Shih et al., 2003). In other words, Web Service is a standard service specification and protocol for dynamic integration between pair-wise systems via the Internet in the form of service.

The Web Service for message delivery on the Internet uses SOAP, while WSDL is used to define the service interface, data type, message format, and other service messages of the Web Service. Users of the Web Service need not to know the language used to write the service program or any details about program variables. They only need to access the service according to the WSDL definition. This structure is helpful for the integration and collaboration of different applications. It is also useful for application developers to release or find required service components via the UDDI registration search mechanism (Liu et al.; Shih et al., 2003).

3. Learning System Structure

In the current market, course material and learning course interfaces that conform to the SCORM standard are all configured to run off a single server (Tung, 2003), as shown in Fig. 2. Users cannot access SCORM course resources on different operating platforms with a single communication protocol. This kind of learning system structure is easier to implement and is suitable for small scale teaching environments. However, for large scale resource servicing users, the required network bandwidth and file accessing service loading will cripple this type of single server learning services. Therefore, when a system is required to service learning courses to multiple users at the same time, there must be multiple digital learning systems to be able to support the large service demands. This will result in another issue of multiple service entrances that confuses users as to which learning system will be able to provide a more efficient service for them. In addition, the LMS must be able to simultaneously provide course search, course download, learning condition history, and necessary course sequence ordering tasks that only high priced servers can afford to handle such massive workloads (Liu et al.; Shih et al., 2003).

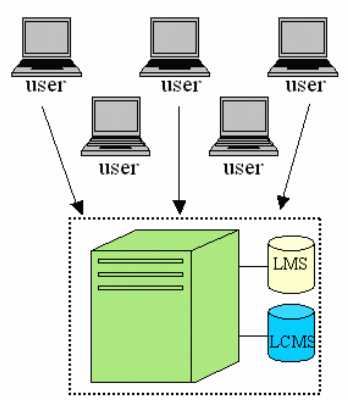

Fig. 2. Traditional Single-Server SCORM Structure

3.1. ***Traditional Decomposed SCORM Learning System Structure***

To improve service efficiency, the learning system is split into two independent sub-systems LMS and LCMS as a realizable solution, as shown in Fig. 3. LMS operates with processes as its basis, and is mainly used to manage student learning information and their learning progress. Then adjust for the material sequence accordingly. LCMS are mainly used to store and manage course information for student use. This structure will achieve the target of system load spreading, but with the same disadvantage of existent expansion flexibility as single servers. Therefore, this traditional decomposed structure cannot resolve network congestion when massive users are online resulting in lowered performance or even interrupted services of the LCMS.

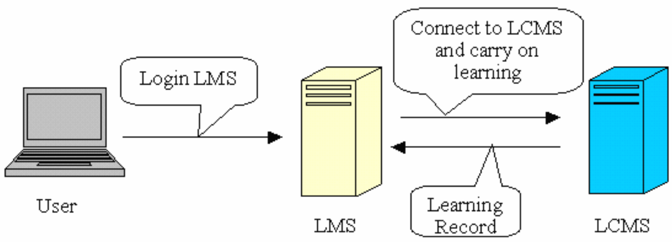

Fig. 3. Traditional Decomposed SCORM Structure

3.2. ***Distributed Learning System with Load- Balancing Function***

The learning system can be split into LMS and LCMS sub-systems that act as independent servers. The integrated load-balancing function of LCMS server enables the realization of multiple LCMS providing service for a distributed learning structure (Kao et al., 2006; Chien et al., 2005). After the user selects a course from the LMS, the LCMS Broker will then assign a light-loading LCMS to the user. The assigned LCMS is determined by analysis of uploaded network load information from each individual LCMS. Then it can provide course material for the user to download its services. This structure realizes load balancing of a network and provides a great amount of users to simultaneously access a course of learning, but the LCMS Broker is generally designed by a higher cost of Server (Kao et al., 2006; Chien et al., 2005). The proposed embedded LCMS Broker not only provides the same performance as higher cost of Server but also obviously decreases the cost of establishing system.

4. The Proposed Decomposed SCORM Structure with Embedded LCMS Broker

To achieve the goal of SCORM for the sharing of teaching materials, this paper provides a Web Service distributing learning management system in conformity to the SCORM standard and capable of integrating learning resources and balancing network traffic. The proposed decomposed SCORM system structure with embedded LCMS Broker is shown in Fig. 4.

The system operates in six procedures as the followings.

(1) The user connects to the LMS from his or her PC, logging into the learning platform and embeds the connection program downloaded from the LMS to the PC.

(2) The user connects the PC to the embedded LCMS Broker via the connection program to explore required teaching materials from different LCMS. The LCMS with the lowest load is then selected from suitable LCMS as the source of the teaching materials.

(3) The learning is implemented by direct connection from the client to the selected LCMS. The user requests connection to the LMS via the selected LCMS as the proxy server and implements learning via the LMS learning interface page in conjunction with the LCMS teaching materials.

(4) Dynamic records are accessed by the load-balancing Embedded LCMS Broker to return the IP address of LCMS servers that has lower network loading. The ActiveX component on user browser then receives the IP address and redirects the connection to the appropriate LCMS.

(5) LCMS begins to send related learning courses to the user computer.

(6) LCMS returns the learning history information back to LMS in order to organize the learning sequence based on the user learning conditions.

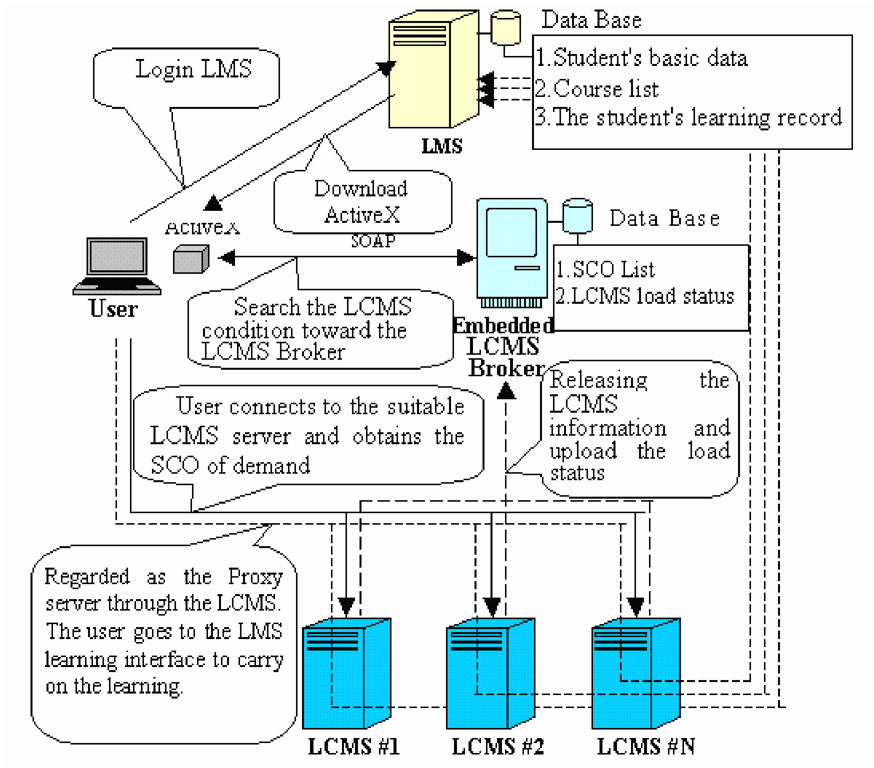

Fig. 4. The Decomposed SCORM Structure with Embedded LCMS Broker

4.1. *LMS / LCMS Structure*

The system has a decomposed structure comprised of the Learning Management System (LMS) for processing basic data, learning data and learning records of learners, and the Learning Content Management System (LCMS) for managing and storing course resources and providing courses to the client (Kao et al., 2006; Chien et al., 2005). The Web Service cross-platform distributing configuration provides common communications between systems and enhances their learning resource integration capability. When logging in the LMS, users may access teaching materials via the LCMS for learning. They only need to log in the LMS for exploration of course resources from LCMS in different regions, indicating an improved teaching material integration capability of the SCORM.

4.1.1. *Load-balancing Capability*

In general, the LMS only needs to interact with the API Adapter of the client and store required data in the system, while the LCMS must provide all teaching materials to the client. The LCMS, therefore, needs more bandwidth and quality hardware for operation than the LMS.

The decomposed structure allows subsystems to be added only to that part of the entire system that has higher network traffic. The load of the LCMS increases when more users access teaching materials from the system at the same time. In these circumstances, adding another LCMS and transferring some of the users to this subsystem for service will improve the performance of the learning platform.

4.1.2. *Combination of Resources and Performance LMS / LCMS Structure*

When users cannot find required teaching materials from the original LCMS, the system will help them to search resources from other LCMS and select the LCMS with the lowest load from suitable LCMS to be the source of the teaching materials. The structure of this study requires only one embedded LCMS Broker. Users may search information on teaching materials from various LCMS on the Internet via this embedded Broker and the integrated Web Service interface.

4.2. *Embedded LCMS Broker*

In order to save the cost of system building, the low-cost embedded load-balancing Broker is used to replace the higher-cost LCMS Broker Server. The improved system structure is shown in Fig. 5. The S3C4510 ARM7TDMI microprocessor developed by Samsung is used here as the core of the development kit (See Fig. 6.) used to design the proposed Embedded Broker with load-balancing capability. The load-balancing algorithm is designed by dynamic recording the load status of LCMS. The software architecture will be based on the uClinux operating system, which supports TCP/IP stacks. The selected microprocessor is a 32-bit RISC embedded processor with the feature of low power consumption (Cui et al., 2005; Teng, 2004). This will be able to implement network applications such as FTP, Telnet, TFTP, NFS and the Load-Balancing previously mentioned.

As current research shows, information on course resources is generally explored via the LMS. To decrease its load, this study provides a structure under which users search information via the embedded LCMS Broker directly. When logging into the LMS for the first time, the client will need to download a connection program from the LMS and embed it in the browser, so that it may access service and explore information directly from the embedded LCMS Broker. The load of the LMS is effectively decreased with the proposed system structure.

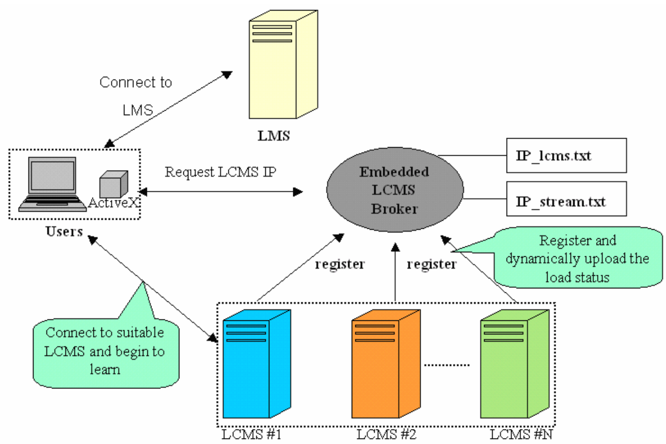

Fig. 5. A Decomposed LMS/LCMS Structure with an Embedded LCMS Broker

Fig. 6. Development Board of the Embedded Broker

4.3. *Cross-Domain Scripting Issue*

Since the SCORM standard integrates teaching materials and platforms using JAVA Script, the cross-domain scripting issue may occur when the learning system and course resources are stored in subsystems of two different domains. Traditionally, the SCORM learning environment located on two different domain servers cannot be integrated into a single structure, as the API Adapter is a component in LMS, while the driving JAVA Script is contained in the teaching material of LCMS. The API Adapter may reject JAVA Script calling of the course for the sake of JAVA security (ADLNet, 2005).

The improved structure of this study no longer uses LMS JAVA Applet as the API Adapter. Instead, the API Adapter is contained in the LCMS for data transmission and embedded in the LMS web page via dynamic hyperlinks. When the user finds the most suitable LCMS via the embedded LCMS Broker, the LCMS becomes a proxy server under the support of URL Rewrite technology. The user requests connection to the LMS via the selected LCMS proxy server and implements learning via the LMS learning interface page in conjunction with the LCMS teaching materials. No connection is required from the client to the LMS in the circumstances and, for the browser, LCMS

JAVA Script is used by users to drive LCMS API Adapter. With the URL Rewrite technology, the API Adapter contained in the LCMS and embedded in the LMS web page via hyperlinks can be accessed by the LMS. The system structure of this study provides an effective solution for the cross-domain scripting issue.

5. Experiment Results

In this section, the performance tests of traditional single-server system structure, traditional decomposed system structure and decomposed system with LCMS Broker would be explored respectively. All response time tests in the study are conducted by analyzing the packet response time, acquired by using Ethereal, a professional program for packet capturing analysis. In order to retrieve the packets communicated between LCMS Broker and each learner's PC for analysis of data transmission, Ethereal is utilized (Network packet analyzer, version 0.10.10). In figure 7, the x-axis stands for the number of learners while the average response time for a traditional single-server system is indicated by the y-axis. While 40 learners are using the system simultaneously, the averaged response time is over 200 seconds and meets the limit of system hardware. For a traditional decomposed structure (with one LMS and one LCMS), the averaged response time appear as a rising trend when the learners gradually increase, as shown in Fig. 8. This means the load of LCMS could not be balanced under the traditional decomposed system structure.

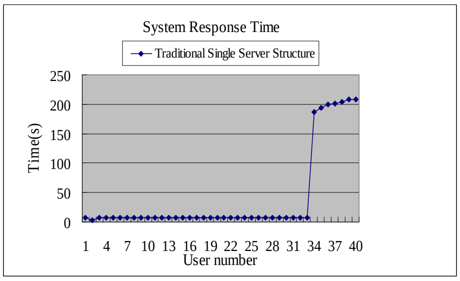

Fig. 7. System response time of single-server structure

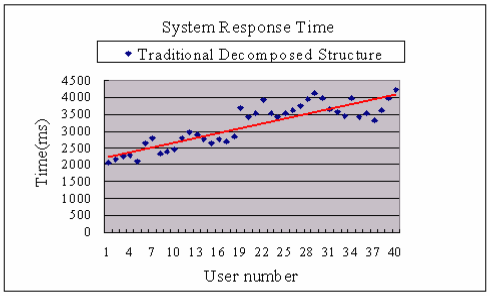

Fig. 8. System response time of a traditional decomposed structure

This paper provides a load-balancing structure with a LCMS Broker that balances LCMS network traffic effectively. First, the implementation of proposed load-balancing system structure with a LCMS Broker designed by a personal computer (PC) is shown in Fig. 9. The structure is capable of creating an integrated SCORM learning environment by collaboration of two servers in different domains to improve the resource integration capability of the SCORM and the response efficiency of the system. The user may log in the LMS to acquire teaching materials and information on the LCMS server, which is selected by the PC Broker. According to the dynamical recording method, the designed PC Broker selects an appropriate LCMS Server to serve the user. Under PC Broker Architecture, the experimental results for the system's response time against increasing the number of learners are shown in Fig. 10. When the users continuously connect to the LCMS Broker to request the IP of most suitable LCMS, the maximum averaged waiting time for the user is about 2.3 seconds, as shown in Fig. 10. This means that the user only needs to spend about 2.3 second at most to connect the appropriate LCMS and begin the learning. Under the same system structure, if the system structure replaces the PC Broker with the proposed embedded LCMS Broker (shown in Fig. 11), the averaged response time is then improved as shown in Fig. 12. The maximum averaged waiting time for the user is about 1.3 seconds at most. The performance comparison between these two system structures using different Brokers is demonstrated in Fig. 13. Therefore, the proposed embedded LCMS Broker is actually able to distribute the loads on servers and improve the operating performance of networks. The performance test results of different system structures are finally gathered together in Table 1. These tests include the results under the proposed system structure with using different load-recording methods (polling method and dynamic recording method). The proposed decomposed structure with embedded LCMS Broker (using dynamic recording algorithm) not only has the best system performance, but also has less cost than that with PC Broker. By the application of embedded LCMS Broker with load-balancing function, the operating performance of the system network is obviously improved, and it is also hoped that this improved network environment can promote the learners' interest and efficiency.

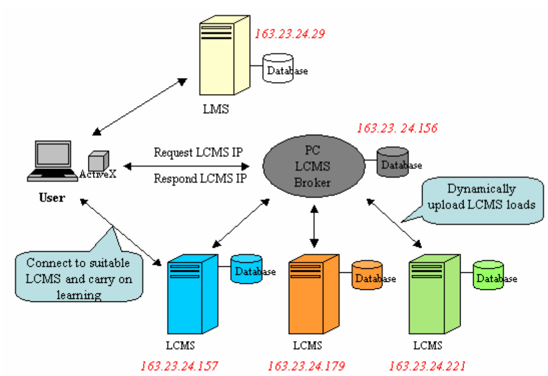

Fig. 9. The proposed system structure with PC Broker

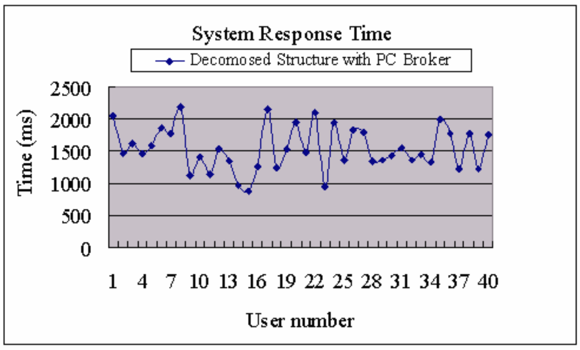

Fig. 10. Response time of decomposed system structure with PC Broker

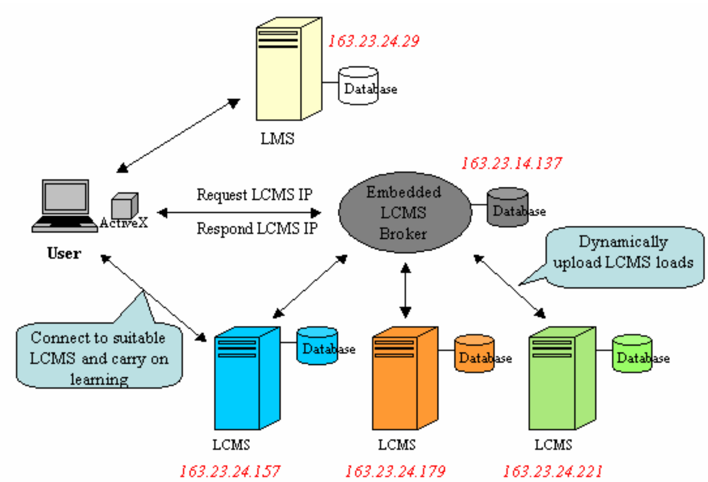

Fig. 11. The proposed system structure with embedded Broker

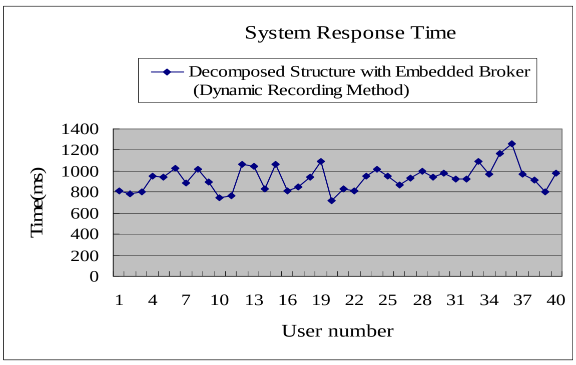

Fig. 12. Response time of proposed system structure with embedded Broker

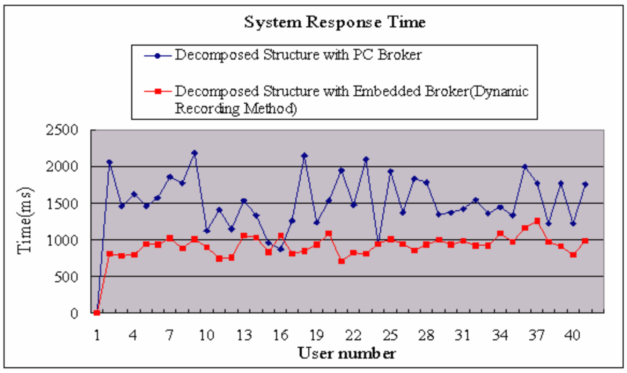

Fig. 13. Comparison of response time between different system structures

Table 1. Response times under different system structures

Structure	Response Time (Second, s)	Efficiency Rank
Decomposed Structure with Embedded Broker (Dynamic Recording Method)	0.7s ~ 1.3s	1
Decomposed Structure with PC Broker	0.9 s ~ 2.3s	2
Traditional Decomposed Structure	2.0s ~ 4.3s	3
Decomposed Structure with Embedded Broker (Polling Method)	5.0s ~ 8.0s	4
Traditional Single Server Structure	8.0s ~ 210s	5

6. Conclusions

Standardization of format solves the problem of teaching material transfer, but courses cannot be shared amongst individual learning platforms. Most current research focuses on the development of functions and teaching materials based upon a single learning platform, rather than the integration of multiple platforms for users to access course resources from a single portal. Even though there is a good dependency between teaching materials and platforms, the user who needs to transfer resources among platforms must import data for individual courses in each platform. The recourse sharing efficiency in the circumstances is, therefore, significantly affected.

In this paper, we propose a decomposed SCORM structure that effectively shares LCMS network traffic by the load-balancing function of embedded LCMS Broker. The proposed embedded LCMS Broker can not only balance the loads between LCMS, but also obviously reduce the cost of hardware. The proposed structure is capable of creating an integrated SCORM learning environment by collaboration of two servers in different domains to improve the resource integration capability of the SCORM and the response efficiency of the system. The system of the study consists of the Learning Management System and Learning Content Management System under the Web Service structure. This decomposition structure is helpful for load balancing of LCMS in multiple domains.

Required teaching materials are accessed from individual LCMS and the LCMS with the lowest load is selected from suitable LCMS as the source of the teaching materials. The user may log in the LMS to acquire teaching materials and information on server load status via the embedded LCMS Broker. When the user select a required course and enters its list, the system will dynamically assign the most appropriate LCMS for service when the user selects the same course in the future. The URL Rewrite technology is used by this study to successfully create a SCORM learning environment on two servers in different domains. This structure not only provides a solution for the cross-domain scripting issue, but also improves the course resource integration capability of the SCORM and the response efficiency of the system to a significant extent.

Acknowledgements

This paper has been supported in part by the grant NSC95-2520-S-212-001-MY3 from the National Science Council of Taiwan.

References

Aakanksha & Punam Bedi, (2007). Load Balancing on Dynamic Network using Mobile Process Groups, 15th International Conference on Advanced Computing and Communication, 553-558.

ADLNet, (2004). SCORM 2004 3RD Edition Sequencing and Navigation (SN) Version 1.0 DRAFT, http://www.adlnet.gov/downloads/290.cfm

Canadian Department of National Defense, (2002), SCORM Dynamic Appearance model, http://www.online-learning.com/papers/SCORMModel.pdf.

Chien, T.C. & Kao, F.C., (2005). The Design of Load-Balancing LMS Based on Decomposition Structure. IEEE International Conference on Advanced Learning Technologies, 783 – 787.

Chu, C.P., Chang, C.P., Yeh,C.W., Yeh, Y.F., (2004). A Web-Service Oriented Framework for building SCORM Compatible Learning Management Systems. International Conference on Information Technology: Coding and Computing, (1), 156-161.

Cui, K. & Wu, Z.W., (2005). Research and implementation of remote monitoring system based on real-time uClinux. Proceedings of ICSSSM '05, (2), 1182-1187.

Dodds, P., et al. (2001). SCORM Content Aggregation Model Version1.2, Advanced Distributed Learning Initiative, http://www.adlnet.org/.

Dodds, P., et al. (2001). SCORM Run Time Environment Version1.2, Advanced Distributed Learning Initiative, http://www.adlnet.org/.

Giannacopoulos, D.D. & Ren, D.Q., (2006). Analysis and Design of Parallel 3D Mesh Refinement Dynamic Load Balancing Algorithm for Finite Element Electromagnetics with Tetrahedra. IEEE Transactions on Magnetic, VOL. 42, NO. 4, 1235-1238.

Hsu, W.J. & Lin, P.C., (2003). Learning Technology Standards and SCORM. Joumal of Library and Information Science 29(1), 15-28.

Itschner, R., Pommerell, C., Rutishauser, M., (1998). Remote monitoring of embedded systems in power engineering. IEEE Internet Computing, 2(3), 46 – 52.

Kao, F.C. & Feng, T.H., (2006). The Design of Decomposed LMS with Embedded 3D Virtual Instruments. The 2006 International Conference on SCORM, 93 – 98.

Liu, P.Y., Hsu, L.H., Chakroborty, A., (2002). Towards Automating the Generation of SCORM-Based Multimedia Product Training Manuals. ICME’2002 conference, 397-400.

Liu, X.F., Saddik, A. E. Saddik, Georgana, N.D., (2003). AN IMPLEMENTABLE ARCHITECTURE OF AN E-LEARNING SYSTEM. CCECE, (2), 717-720.

Mccombie, B., (1998). Embedded web server now and in the future. Real-time Magazine. No.1, 82-83.

Nehra, N. & Patel R.B., (2007). Towards Dynamic Load Balancing in Heterogeneous Cluster using Mobile Agent. International Conference on Computational Intelligence and Multimedia Applications, 15-21.

Tresso,T., (2002). An automatic evaluation system for technical education at the University level. IEEE Transactions on Education, 268-275.

Tung, T.H., (2002). Apply Knowledge Map to Develop Physics Problem-Solving System. Thesis of Master, Dep. Information and Computer Engineering, Chung Yuan Chrisian University, Chung-Li, Taiwan, R.O.C.

Shih, T.K., Chang, W.C., Lin, N.H., Lin, L.H., Hsu, H.H., Hsieh, C.T., (2003). Using SOAP and .NET Web Service to Build SCORM RTE and LMS. 17th international Conference on Advanced information Networking and Applications. 408-418.

Teng, J.H., Tseng, C.Y., Chen, Y.H., Kim, S.H., Hong, J.W.K., (2004). Integration of networked embedded systems into power equipment remote control and monitoring. TENCON 2004, C (21-24), 566 – 569.

Wang, S.C., Wang, W.C., Yu, W.H., Chen, C.S., Hsieh, S.H., (2002). A Study on E-Teaching/E-Learning Standardization and Courseware Sharing. Journal of Engineering, (85), published by Taiwan University.

Wilson, A, the challenge of embedded internet design, Real-Time Magazine, No.1, March 1998, pp.78-80.

Yang, J.T.D., Tsai, C.Y., Lin, T. et al., (2003). A SCORM-compliant Content Repository for Sharable Learning Objects. WISCS 2003, 25-30.

Zhu, Y. & Hu. Y., (2003). Efficient, Proximity-Aware Load Balancing for Structured P2P Systems. Proceedings of the Third International Concerence on Peer-to-Peer Computing(P2P'03).