Designing an Open Architecture of

Agent-based Virtual Experiment Environment on WWW

Chi-Wei Huang, Chang-Kai Hsu, Maiga Chang and Jia-Sheng Heh
Department of Information and Computer Engineering, Chung Yuan Christian University
Chung Li, 32023, Taiwan, Tel: 886-3-4563171 ext. 4725
E-mail: jsheh@ice.cycu.edu.tw

Abstract: This paper proposes an open architecture of *virtual experiment environment* on WWW by using intelligent agent techniques. Several experiment components and a science experiment agent are analyzed and implemented to support teachers to quickly assemble a social learning system. Such design is based on distributed network architecture to attain high reliability and efficiency.

This architecture of agent-based experiment environment is composed of two major layers, *Coordinate Layer* and *Communication Layer*. And KQML (Knowledge Query and Manipulation Language) is used to realize communication protocols and information sharing language for the interactions among experimental equipment and science experiment agents. A workable experiment system based on our architecture, called *SoftLab*, is implemented on WWW to demonstrate its feasibility and flexibility.

Keywords: *virtual experiment, experimental equipment, WWW (World Wide Web), KQML (Knowledge Query and Manipulation Language), intelligent agent*

Introduction

In the last ten years, personal computers have sparked a revolution in education. Everyone from preschool children to senior citizens can now develop science projects and prepare reports by using computer technology at home. Besides being passive learning tools as textbooks, paper and pens, computer-aided education (CAE) systems can prompt learners some feedback and responses in appropriate ways during learning process. (Diessel 1992; Futtersack 1992; T.-W. Chan 1995) Take interactive tutorials for example, this kind of CAE systems can not only teach learners specific knowledge, but also evaluate what learners has been learned and re-teach based on the evaluation. Any paragraph in a course of study can certainly be reviewed to meet the needs of learners. (J.C. Gonzalez 1992; H. Lianjing 1992)

Considering the practicum of conventional education, "learning by doing" design brings the idea of situated learning into computer environment, as the interactions of human cognition are studied and improved in different computer-based learning environment. (Brown, Collins & Duguid, 1989) Hence, a simulated platform which provides learners experiment with visual equipment on computers, called *virtual experiment environment* (VEE) had been constructed on the computer network. (Y.-W. Jeng *et al*. 1996 J.C. Gonzalez *et al*. 1998) This paper proposes an open architecture of agent-based *virtual experiment environment* on WWW.

Section 2 identifies these problems when establish a *virtual experiment environment* on WWW. Besides several important issues are analyzed, an open architecture then can be constructed. After the open architecture is proposed, more detailed designing for a specific agent-based architecture are designed by Section 3. An experiment system constructed on WWW, is so-called *SoftLab*, providing a workable agent-based platform of physics in Section 4. Section 5 makes a summary and discusses those possible future works.

Architecture of Agent-Based Virtual Experiment Environment

The great advantages of WWW are information transmission and knowledge sharing. Unfortunately, those capabilities to support high quality interactive learning, retaining information of learners and even tracking the learning progress are still under disputed. Since an open architecture of agent-based *virtual experiment environment* on WWW is proposed, the idea of Software ICs (Integrated Circuits) should be taken into consideration.

Based on the precompiled binary code, *components* become independent of the language in which they were created. Furthermore, there are two more characteristics of components: reusability and flexibility. *Reusability* means that components can be used in many applications, continually enhancing and improving without having to recompile. On the other hand, the *flexibility* of components provides that suitable components can be simply chosen to extend the standard functionality for the specific needs.

Although the Software ICs are very useful for establishing such open architecture on WWW, there must have a container comprised several ICs and channels for exchanging data. It implies that if some experimental equipment used for experiments, an experiment environment must exist firstly. And than the experiment can be observed through those interactions in experiment environment. According to such viewpoints, a uniform interface among components is also necessarily provided by the open architecture.

Cooperative learning can improve both professional and social skills of learners, including face to different opinions and interdependence of group members. (Jamie C *et al*. 1998; Chanchai Singhanayok 1998; Doug L 1998) Therefore, the communication mechanism between participants of experiments has to design is also an important issue in order to achieve the goal of constructing synchronized and cooperative experiment environment.
As mentioned above, all issues are taken into consideration. The architecture of agent-based virtual experiment is then constructed as (Fig. 1) shown.

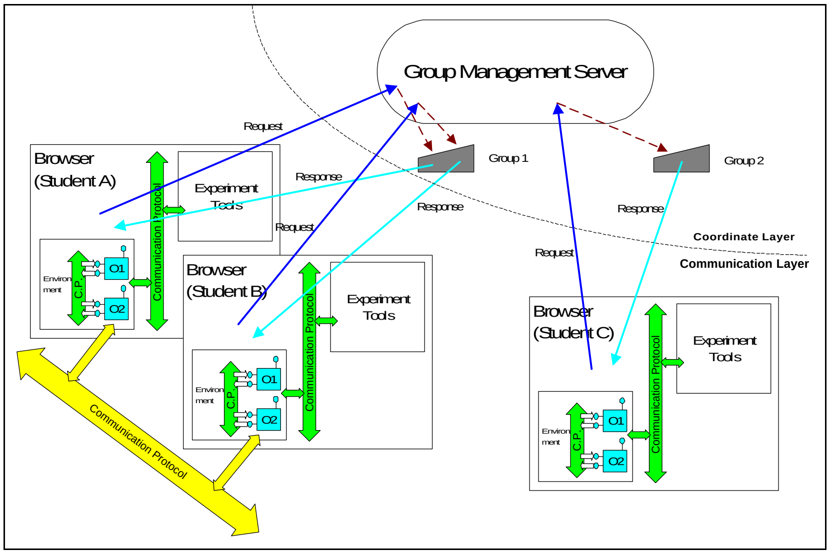

Figure 1. An open architecture of agent-based virtual experiment

(Fig. 1) indicates the architecture is composed of two major layers, Coordinate Layer and Communication Layer. Because distance learners may use browsers to participate in the collaborative online activities, the responsibility of *Coordinate Layer* is to deal with the cooperative learning when experiment in the virtual environment through the computer networks. And learners can work together for accomplishing a common task or experiment through the agency of *Coordinate Layer*.

Besides these interactions among participants, interactions among components in the *virtual experiment environment* have to handle. To describe and maintain this sort of relations between the equipment, such like ball and slope in the experiment, a management manner as *Communication Layer* provides is necessary.

Generally speaking, there are some standard physics laws illustrating the relationships between equipment (or simulated objects) in the environment to regulate the interactions. Two-ball collision is a good example for explaining. The key point is that balls never know they have a collision with the other one. They only know that if they are forced, then they must response an action corresponding to the excitation. So the environment must provide a mechanism to coordinate the behaviors of the objects and transfer the responses appropriately.

Although an open architecture with agent-based virtual experiment on WWW is constructed in this section, more detailed designs still need be done before this architecture can really open for educational purposes. After the *Communication Layer* is well designed, either learner to cooperatively /competitively learn with learners or teacher to teach/evaluate/answer learners will be replaced by intelligent agents (IAs).

Designing the Virtual Experiment Environment

Besides the utilization of component technology, more important issue which makes the architecture constructed in the previous section becomes open is a graph model for representing the relationships between experimental equipment. The detailed explanation for the graph representation is in the following.

Actually, experimental equipment in VEE must be embedded in such an experiment board in the environment so that these interactive relations between experimental equipment can be generated based on the interaction principles. There is three-stage mechanism in the experiment board to make the interactions between experimental equipment come true. At first, the relations between experimental equipment are described, and a directed graph is adopted to describe the experiment and store it in the form of adjacency-list. Secondly, the standard physics laws are pre-stored in the experiment board. It means each node in the adjacency-list will be searched, and a mapping from experimental equipment into the desired physics laws. The last stage implements the experiment based on the standard physics laws and those data structure produced in the second stage.

The relations between experimental equipment are described as a *directed graph* $G(V, E)$, where V is a finite set and E is a binary relation on V. The *vertex set* V of G contains all the *experiment equipment vertices* in this experiment; whereas, the *edge set* E of G stores all the *relation between the equipment*. (Fig. 2a) represents an example directed graph G of vertex set {1, 2, 3, 4}. If (u, v) is an edge in graph G, then the equipment u will interact with equipment v. An adjacency-list of (Fig. 2b) stores the relations between the equipment vertices in (Fig. 2a).

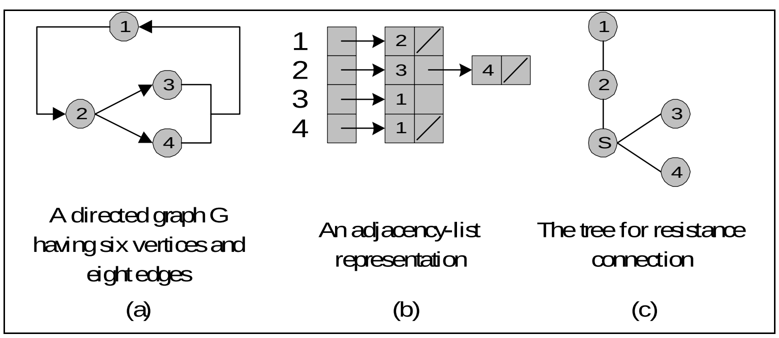

Figure 2. The three-stage mechanism

Take electric resistance connections for example to illustrate the above three-stage mechanism. The electric experimental equipment in (Fig. 3) is added as battery, resistance 1, 2 and 3. From (Fig. 2a) the representation of experiment after transformed into graph is shown in (Fig. 3), which means graph G has four vertices: V_1-Battery, V_2-Resistance 1, V_3-Resistance 2, V_4-Resistance 3, and five edges: (V_1, V_2), (V_2, V_3), (V_2, V_4), (V_3, V_1), (V_4, V_1).

After the first stage, the experimental graph then will be stored in the form of adjacency-list as (Fig. 2b) shown. Finally, a specific tree is generated from the adjacency-list based on the standard physics laws just like (Fig. 2c) shown. Accordingly, all the physics quantities in the electric circuit of this experiment can then be obtained.

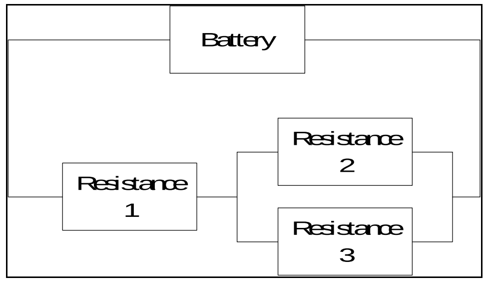

Figure 3. Electrics resistance connections example

Regarding the agent-based issue, intelligent agent's behaviors for the educational proposes should be analyzed first. Three behaviors can be simply told out as follows:
1) Online recording,
2) Online replaying,
3) Online navigating.
In most cooperative learning systems, it is necessary to design a communication protocol for information and knowledge exchange. In the open architecture of *agent-based virtual experiment environment*, the most useful agent communication language, *Knowledge Query and Manipulation Language* (KQML), is chosen as our communication protocol in *Communication Layer*.

KQML focuses on an extensible set of performatives, which define the permissible operations that agents may attempt on each other's knowledge and goal stores. The performatives comprise a substrate on which to develop higher-level models of inter-agent interaction such as Contract Nets and Negotiation. (Tim Finin 1992)

Conceptually, a KQML message consists of a performative, its associated arguments which include the real content of the message, and a set of optional arguments which describe the content in a manner which is independent of the syntax of the content language. For example, a login to a group learning system might be encoded as:

(*tell* :*content* (Login (Ken (jk1234y, user)))
:*language* list
:*ontology* Group-Learning)

In this message, the KQML performative is *tell*, the *content* is (Login (Ken (jk1234y, user))) and the assumed *ontology* is identified by the token :Group-Learning.

Implementing the Virtual Experiment Environment on WWW

(Fig. 4) in the following depicts the architecture for electricity, only with minor differences from the original one. An experiment system of physics course then is implemented under the open architecture proposed, called *SoftLab*. Besides these synchronous and asynchronous tools for Web-based learning, an intelligent science experiment agent, called *Lancelot*, is also added in *SoftLab*. The specific science experiment agent owns its architecture just like (Fig. 5) indicates. Distance learners are allowed to participate in those collaborative on-line activities through browsers. And with the characteristics of Web technology, it even can increase students' awareness of related concepts through the inclusion of hypertext links to other Web-based information. Chat-room is the only component that can be used by learners to communicate with others in VEE. Teacher, moreover, can also use it to give hints and comments.

Considering the benefits of component-oriented analysis, Microsoft ActiveX is chosen to implement the experiment system described above. It is easy to divide the whole system into components with abstract models of abilities and behaviors. Each learning work is implemented as an independent component. Owing to ActiveX technology, Microsoft Internet Explorer is chosen as the platform of VEE. And Visual Basic Script is then the bridge between components.

When designing the experiment system, performance issue of the system is the most important. The traditional BBS (Bulletin Board System) server is analyzed and modified in order to solve this problem. The major modification objectives include discussion board, flow control of learning, privilege management and group management. There are three parts in the whole *SoftLab*, including VEE, experiment tools and the science experiment agent Lancelot, as (Fig. 6) shown.

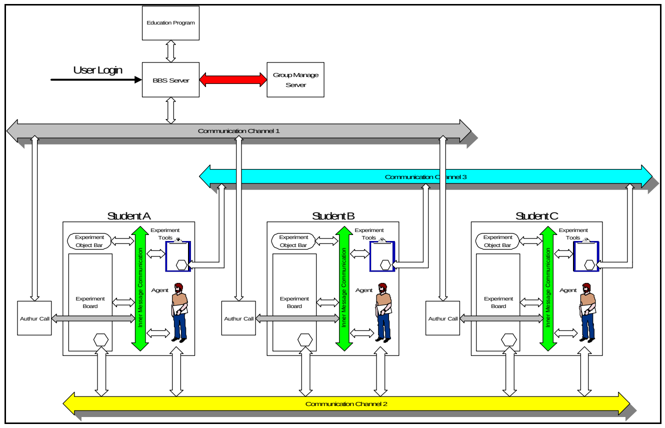

Figure 4. Architecture of *SoftLab*

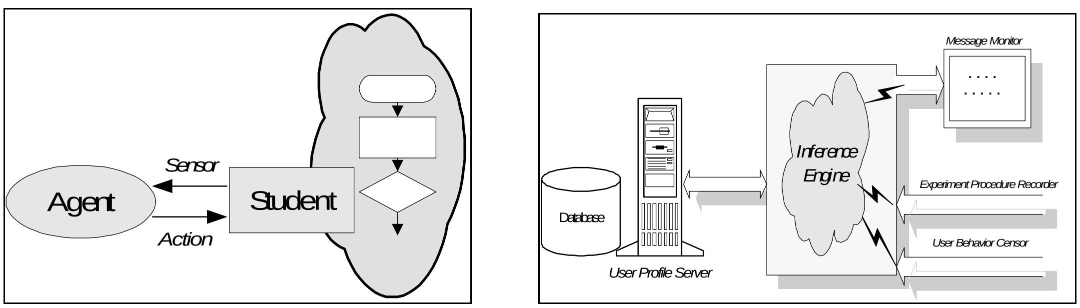

Figure 5. Architecture of science experiment agent

Figure 6. *Virtual experiment environment* of circuit experiment

Conclusion

In this paper, the possibility of constructing an open architecture of agent-based VEE on WWW is fully discussed and analyzed. Two major layers compose the architecture of agent-based experiment environment that are *Coordinate Layer* and *Communication Layer*. And the interactions between *experimental equipment* are realized by KQML. *SoftLab* is a workable experiment system implemented on WWW based on the architecture proposed in this paper to demonstrate the feasibility and flexibility.

Another interested research direction is the courseware and visual components design when an interactive user interface is used. As mentioned in the introduction, we believe that only those learning workspace systems designed with open architecture, such as VEE, can be extended easily. Furthermore, an intelligent agent system approach, such as the mechanism of knowledge formation during learning, might be analyzed deeply. In the agent-based VEE, learners can process a *virtual experiment* with agent cooperatively/competitively. Hence, how to evaluate the learning progress of learners is also a good research topic.

Reference

Chi-Wei Huang, Shiao-Ting Sun, Chia-Chin Chang, Kuo-Chang Jan and Kun-Yuan Yang. (1998). Implementing a Science Experiment Platform integrated with Real Experiment on WWW. 2nd Global Chinese Conference on Computer in Education, Hong Kong, 1998, pp.50-57

T.W. Chan. (1996). A Tutorial on Social Learning Systems. Emerging Computer Technologies in Education, AACE, 1996

Yu-Wei Jeng, Maiga Chang, Ivory Chung and Jia-Sheng Heh. Designing Objects for Virtual Experiments. OOTA'96, Taiwan, 1996, pp.331-339

Th. Diessel, A. Lehman. An ITS for engineering domains: concept, design and application. 4th International Conference, ICCAL'92 Wolfville, Nova Scotia, Canada, June 17-20, 1992 Proceedings

M. Futtersack, J.-M. Labat. QUIZ: A distributed intelligent tutoring system. 4th International Conference, ICCAL'92 Wolfville, Nova Scotia, Canada, June 17-20, 1992 Proceedings

J.C. Gonzalez, J.J Sancho, J.M. Carbo, A. Patak, F. Sanz. Intelligent tutoring system in medicine through an interactive. 4th International Conference, ICCAL'92 Wolfville, Nova Scotia, Canada, June 17-20, 1992 Proceedings

H. Lianjing. A tool for developing intelligent tutoring system. 4th International Conference, ICCAL'92 Wolfville, Nova Scotia, Canada, June 17-20, 1992 Proceedings

Jamie C. Cavalier and James D. Klein. Effects of Cooperative Versus Individual Learning and Orienting Activities During Computer-based Instruction. ETR&D, Vol. 46, No. 1, 1998, pp.5-17

Chanchai Singhanayok and Simon Hooper. The Effects of Cooperative Learning and Learner Control on Students' Achievement, Option Selections, and Attitudes. ETR&D, Vol. 46, No. 2, 1998, pp.17-33

Doug L. Maskell and Peter J. Grabau. A Multidisciplinary Cooperative Problem-Based Learning Approach to Embedded Systems Design. IEEE Transactions on Education, Vol. 41, No. 2, May 1998, pp.101-103

Tim Finin, Rich Fritzson, and Don McKay et. al. An overview of KQML: A knowledge query and manipulation language. Technical report, Department of Computer Science, University of Maryland Baltimore County, 1992.