Intelligent Multimedia Educational System on Distributed Environment

이세훈, 왕창종

인하대학교 전자계산공학과

요 약

본 연구는 분산 환경에서 지능형 교육을 지원하는 멀티미디어 교육 시스템을 제안한다. 제안된 시스템은 개별 학습과 실시간 사용자 상호작용을 지원할 수 있게 설계되었다. 본 시스템을 CORBA 기반으로 설계함으로써 다중 사용자 환경에서의 그룹 관리를 위한 모듈을 두어 효율적인 분산 처리가 가능하게 하였으며, MHEG 표준을 사용함으로써 멀티미디어 코스웨어와 실시간 사용자 상호작용을 지원한다. 또한, 학생의 응답을 진단하고 평가를 생성하기 위해서 언어 변수 개념과 퍼지 이론을 사용하였다.
본 시스템은 문제의 자동 생성과 동적인 교수 전략을 제공할 수 있으며, 멀티미디어 표현 방식과 상호 교환 방식에 있어서 표준을 채택함으로써 코스웨어의 재사용성이 우수하다. 또한, HTTP 의 수동적인 프로토콜과 상호작용성의 부족을 극복하기 위하여 CORBA 와 MHEG 을 사용하였다.

Abstract

This paper suggests a multimedia educational system which has the ability to extract intelligent instruction on the distributed environment. The proposed system is designed for supporting individual instruction and real time user interaction. As the system based on CORBA, we put group managing module on it for multi user environment, so it has ability for distributed computing facilities. Using MHEG standard, we can provide multimedia courseware and real time user interaction. To diagnose students' responses and generate evaluations, we use several linguistic variables and fuzzy theory.
There are two major advantages for using this system. This system can provide dynamic generation of problems and the ability to provide a dynamic instruction strategy. And it can increase reusability of courseware material for using standard of multimedia representation and communication. We use CORBA and MHEG to overcome the disadvantage of the Web, passive protocol and poor interactivity, HTTP.

1. Introduction

The Web has been spread world wide as a standard of distributing multimedia/hypermedia information. There are many applications, include distance learning, based on the Web[Kap91, Nak95]. The advantage of using distance learning on the Web is apparent. We can provide instructional information using a hypertext form, the same as students' cognitive structure[Cra98, Dwy95, Ibr95, Car95]. But there are some disadvantages using the Web: passive protocol and poor interactivity[Bol98]. These problems can be overcome with two models, that could be integrated in WWW. A uniform paradigm to encapsulate and structure web components can be achieved thanks to the OMG's CORBA[OMG95]. And real time user interaction facilities can be achieved by international standard, MHEG[ISO95].
Despite the WWW is able to access distributed information and services on Internet, it can not provide individualized information to students[Bol98]. In addition, unlike existing stand-alone ITS, students must have the ability to study freely and search related information easily. Therefore the system must overcome potential disorientation problems that could burden accessibility.
In this paper, we suggest an intelligent multimedia educational system that can make intelligent instruction and individual instruction on distributed environment. Our system can support concurrent learning and dynamic lesson planning for each student by using linguistic variables[Yoo94] and fuzzy theory. In addition, our system provides

instructional units in which students’ response are diagnosed, and compared, thereby allowing dynamic instructional strategies to develop in step with the students’ progress.

2. Consideration for Multimedia Educational Systems

In this chapter, we consider major problems for developing multimedia educational system on the Web.

2.1 Problems of the Web as educational infrastructure

We have sometimes asked this question. It comes from the difference between learning and information. Students do not require only delivering information, but learning altogether. The major problem when developing educational system on the Web, is very weak interactivity. For students to learn more effectively, they need a high degree of interaction. But most current Web material is represented as HTML, so the Web today provides only low level interaction[Bol98].

The second problem is poor ability for representing materials[Bar98, Mit98]. Because the Internet depends on HTML, the materials seem to poorly designed books with hypertext capabilities[Bol98].

To increase interactivities and multimedia facilities, there have been many studies such as Java, CGI, ActiveX, and some extensions of HTML. But these methods are not fit for educational requirements. Most students need individual learning only for themselves. When considering difference among students, this is very important. But the Web lacks capabilities for individual learning[Jon95].

2.2 The MHEG standard

The MHEG standard provides representation of multimedia objects on heterogeneous networked environment[ISO 95]. It can support real time user interaction when presentation. Multimedia information, which encoded by the MHEG standard, is the MHEG object. As using MHEG objects as inherited form in applications, we require an environment to present these objects. The MHEG standard defined such environment as “engine” [ISO95].

The MHEG-5 standard is the fifth subset of the MHEG standard. It defines some classes in detail. Those classes are appropriate to some applications such as video on demand, audio on demand, interactive TV and hypermedia navigation [ISO96].

So, we apply MHEG, the standard of multimedia information representation, when developing multimedia educational system on the Web.

3. System Design

In this chapter, we design multimedia system that has intelligent educational capabilities. Figure 1 shows the system configuration.

When learning, students execute web browser that is able to run Java applet. Through log in process, courseware server sends materials to students in forms of MHEG objects. Client browser, include MHEG engine, presents those materials to them. The system logs student’s learning history into student module, and diagnoses them in expert module, and decides student’s next course in instructional strategy module.

When students participate in synchronous education program, such as video lecturing, session management module monitors students’ activities. Event module processes happened user events, and interact with educational modules include student module.

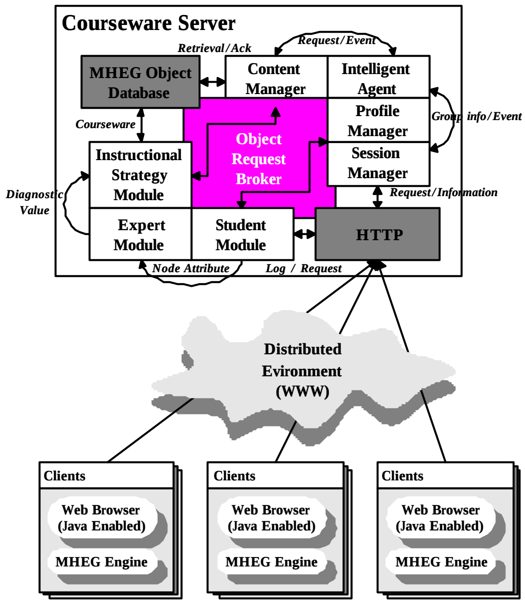

Figure 1. System configuration

3.1 Educational Modules

Educational modules consist of expert module, student module, and instructional strategy module.

(1) Expert module

The domain knowledge will be used in coaching the students, diagnosing possible misconceptions after a student has made an error, and providing appropriate remedial materials. This domain model has dual representation network, the IUN(Instructional Unit Network) and the CN(Conceptual Network).

Within the IUN, each node represents a learning unit, and it includes the theme of units, learning aim, the link of the concepts related to the learning units, the links to the problems, and the links to the subordinate units.

There are two types of the instructional unit. One is the explain unit, the other is the problem unit. The explain unit has instructional materials that have various media types such as text, graphic, still image, sound, and moving picture. The problem unit has two major concept to diagnosing the student's state, difficulty and importance. These concepts are used when lesson planning and gradual instruction objective creating.

The CN is a knowledge network that piles up the knowledge of being contained in subject domain to be learned by the student. The CN is a data structure, which is a set of collected knowledge concerned with the declaring concept of domain knowledge, but independent upon the instructional knowledge. Each node within the CN represents the instructional concept, and the links between the concepts indicate the relationship of them. Using these relations, it is possible to generate problem dynamically, and to diagnose the student's error.

According to environments of the CORBA, expert module can access CORBA ORB to assist easy retrieval of materials, and URL indexing and caching demon to improve performance of system.

Unlike existing stand-alone ITS, user verification have to be supported so as to provide the individual learning according to the rate of each student.

(2) Student module

The instructional strategy module uses the diagnostic result generated by student module in order to inference the basis for determination of learning progress, supplying advice, and selecting comments. In this paper, the student module is based on the overlay model. The diagnostic results of the student, problem type, and examples are saved at nodes in the student knowledge network. And the form of student knowledge network represents the learning path.

When the domain knowledge expert constructs the IUN and the CN by means of authoring interface, the two networks contain slots represented by linguistic variables for holding importance and difficulty that are the attributes of each problem and instructional unit. The linguistic variables used in input are shown in Table 1.

Table 1: Linguistic Variable for Diagnosis

	Instructional unit	Diagnosis result
difficulty	D, M, E	G, M, B
importance	I, M, Ni	
modifier	[Very]*, [Less]*	

* where D = Difficult, M = Medium, E = Easy,
I = Important, Ni = Not Important, G = Good, B = Bad

The linguistic variables about the problem are used for diagnosing student's response. Using the linguistic variables, the domain expert can input at more user-friend environment because of being able to use linguistic variable and no need to decide special real number for attributes of all learning elements. The linguistic variable is the kernel of fuzzy theory that makes use of the fuzzy representation of human being in computer. Using the concentration and dilation in the fuzzy theory, we can define the function that diagnoses the student's current knowledge. The function of concentration and dilation are as follows.

$$CON(A) = A^2 = \sum_{i=1}^{n} \{x_i, m_A(x_i)^2\}, \forall x_i \subset U$$

$$DIL(A) = A^{\frac{1}{2}} = \sum_{i=1}^{n} \{x_i, m_A(x_i)^{\frac{1}{2}}\}, \forall x_i \subset U$$

The author can change the membership functions upon “Good”, “Medium”, and “Bad”, and the diagnostic results can also vary according to the membership function applied. Hence, the author can control the diagnostic results by changing these three membership functions.
The difficulty and the importance of [Table 1] are defined as the dilation of “Medium”(the default membership function of M is M=x), and both “Easy” and “Not Important” are defined as the concentration of Medium.

$$D = I = M^{\frac{1}{2}} = DIL(M)$$

$$E = Ni = M^2 = CON(M)$$

Therefore, we can define the DV(Diagnostic Value) function as follows.

$$DV(R_{1-n}) = norm\left(\sum_{i=1}^{n} (R_c/R_n)^k\right)$$

norm : normalization function that makes the result in the range of [0,1]
R1-n : The problem from start to n that was responded by student.
Rc : The number of correct answer according to linguistic variable
Rn : Total number of problems presented to student
k : The number of powers according to linguistic variable, defined as follows: I or D : ½, M : 1, Ni or E : 2

Normalization is defined as follows: suppose that all problem is correct, then, calculate the DV and divide into the result of calculation about correct answer. If the importance and the difficulty are applied at the same time to the same problem, select one that minimizes the membership degree and applies it.
To use more various values for representing the importance or the difficulty, we must consider modifiers applicable to the linguistic variables. In this paper we define two modifiers, “Very” and “Little”, as follows.

$$V^n D = M^{\frac{1}{2^{n+1}}},\ L^n D = M^{\frac{1}{\sqrt[n+1]{2}}},\ V^n E = M^{2^{n+1}},\ L^n E = M^{\sqrt[n+1]{2}}$$

The equations above always hold the relation

$$V^n E \le L^n E \le M \le L^n D \le V^n D \quad (n \geq 0)$$

regardless of its number of repetition. Hence, the inconsistency of diagnostic result caused by repeated applications of the modifier is removed. For example, “Easy” is never less than “Very Easy” in spite of the number of “Little” modifier's applications.
The calculation result by DV function for the diagnosis of student's response is a real number that ranges between zero and one. To execute fuzzy inference using linguistic variable, result obtained by the DV function has to be fuzzified. Fuzzified result is one of the linguistic variable that consists of Good, Medium, Bad, and the modifier [Very]*, [Less]*.

If we fuzzify the diagnostic results into the linguistic variables, then they can be used together with the inference rules using the linguistic variables. Figure 2 represents the district graph of diagnostic value to fuzzify diagnosed result.

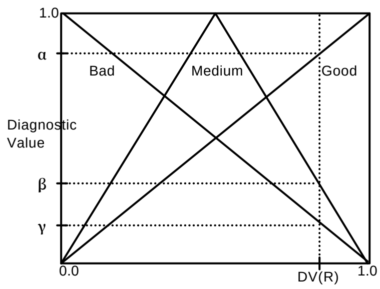

Figure 2 District graph of DV

If the author input the inference rule and there was conflicting among those rules, that is, different result is input at the same condition, the author is presented the condition and is asked what result should be used. The result of learning, whether well or poor, is determined according to what deterministic variable is input by the author. This shows that the author's intention can be reflected to inference rule.

(3) Instructional strategy module

The instructional strategy module mediates and develops the learning of the students through the planning of a curriculum to order the domain knowledge that must be supplied to students, and the planning of teaching to decide the presentation method, re-planning to construct proper planning and monitor the action of students.

Among the contents of a lesson which are taught to the students, the total contents of a lesson are composed of the network of the main objectives of instructions that are relatively independent each other.

Lesson plan presents the evaluation problem and the instructional material of instructional unit, decided by the curriculum planning to students in the proper level. The Instructional strategy module offers students the instructional material on the instruction unit and the evaluation problem by means of the method divided by the curriculum planning.

The gradual instruction objective must be created by considering the student level that has acquired the degree of difficulty of instruction unit and students' records obtained previously. For an efficient instruction, problem presentation order must be planned dynamically by according to the students' level. The instructional strategy module can create dynamically the problem about the concept learning and the discrimination learning using the CN.

Also instructional strategy module can dynamically limit the number of links which the student can select, according to the level of each student. This module have functions which provide the students for advice when the students refer the irrelevant to the present learning content frequently or when disorientation problem occurs.

3.2 Activity Managing Modules

In this section, we design activity managing module for providing time dependent group activities for synchronous learning/training.

(1) Session Manager

Session manager operates as gateway. It authenticates using information in student profile manager. And it manages group participated by students and lecturer. Event handler processes events that occur in each group. Group managing module creates appropriate group for user requirement, and manages those using group pool. Once event occurs, interaction handler catches that. If it cannot be handled, interaction handler sends it to event handler. Event handler processed that using group managing interfaces. Events handled by event handler require group-related operations such as create, destroy, join, and leave. Table 2 defines operations for it.

Table 2 Synchronous learning/training operations

Operations	Functions
Basic Operations	Create Group
	Destroy Group
Group Management Operations	Join Group
	Leave Group
	Manage Applications

	Manage Time Event
	Control Token
	Manage Group Context
Data Transmission Operation	Support Unicast/Multicast
Information Management Operation	Manage and Retrieve User/Group Information

Figure 3 stands out session manager configuration.

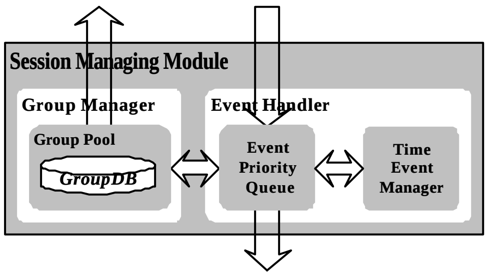

Figure 3. Sesssion Manager

Group manager creates groups as response for user requests, and manages them into group pool. Events caused at each group are sent to event handler, and processed using group managing interface.

Information sent to event handler consists of event data and consumer information. This information is managed as FIFO structure. Time event manager provides time dependent group activities. The creator can set lifetime of group, in this case, a group whose lifetime ended closed automatically. This facility is usable when lecturing has time dependency in educational system.

Time event manager manages information per group unit. Time event service process time event defined by user. When predefined event caused, it send it to group manager.

Group manager manages group context that consists of students information, operates for interaction among students. Group context includes student information who participate group currently, lecturer's information. Group manager maintains context recently processed event. When new event, such as participate and leave, caused context switching to appropriate context. When context switching caused, it notifies students of that case using event handler.

User can play a role of "Group Member" or "Inspector" and lecturer has a "Group Chairman" role. Lecturer as group creator has some privileges. Lecturer can give roles to students, close group oppressively, modify and manage group information.

A student must own token when transmit data to other students. When a lecturer requires token, group manager give token to lecturer immediately. But when a student asks token, group manager sends that request to the lecturer and then the lecturer arranges that. During context switching, such as changing token owner, participate or leave, group manager sends that to event manager.

Group manager provides "SendMessageToAll" operation when a student sends message to all participants. Session manager invokes it and group manager sends real data consists of student identifier and message contents using event handler.

(2) Content Manager

Content manager manages learning contents using workflow concepts. Workflow based data management presents efficient routing of work from person to person or to groups and easy access to pertinent documents. Task is defined students and authors who use contents. Also, Detailed reports of all activities associated with a process. This manager has task manager and flow manager. Author designs the flow and task of learning contents. Task manager and flow manager control the learning content and monitor actions which is performed to contents. Authors and instructors design flow. And course manager maintains components composed in document and relationship among the components. The relationships are made up with update and delete relation. [Fig 4] shows Flow, Task, Task Manager object.

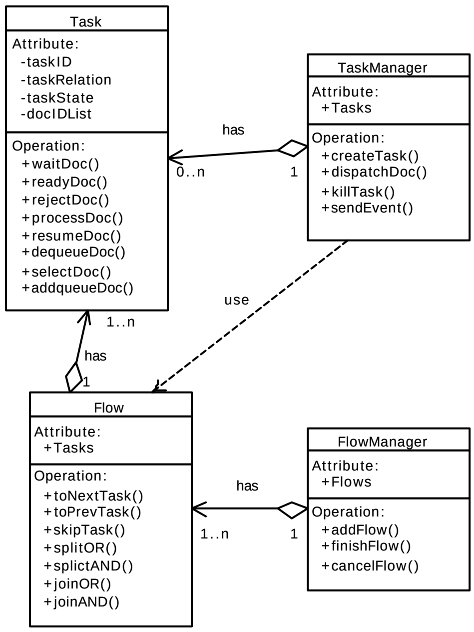

Figure 4. Objects of contet manager

For interface with database management systems, it uses persistence service in Common Object Specification Services (COSS). If we construct cyber university using the framework, this manager is connected to content management center. [Fig. 5] shows Course manager.

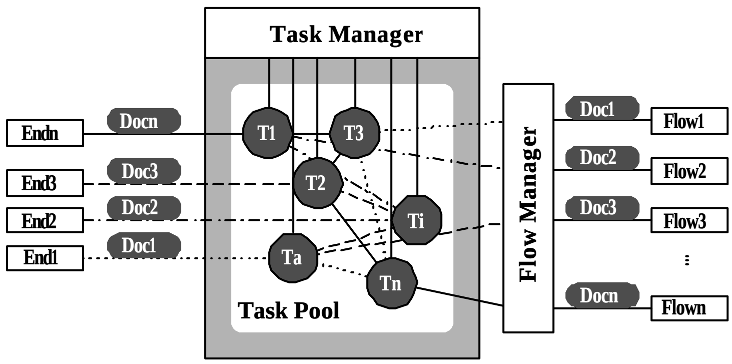

figure 5. Course manager

Content management center helps content professionals-creators, editors and site administrators- manage the full life cycle of content.

(3) Student Profile Manager

Student profile manager maintains student profile, including student private information, result of learning, learning tendency. Profile manager presents operations described in Table 3.

Table 3 Operations of Student Profile manager

Operations	Functions
Basic Operations	Create Users
	Destroy Users
	Fine users
	Modify user profiles

	Registration user's goal
Maintenance of Learning State	GetLearnigHistory
	SetLearningHistory

(4) Intelligent Agent

Intelligent agent determines student's learning course using requirements. Knowledge base is constructed by instructors. Then, determined result is used for designing process of flow of contents in Content manager. Intelligent agent has extract module and diagnosis module. Extract module extracts parameters for diagnosis of student from result of student's learning and return to diagnosis module. [Fig.6] is IDL(Interface Definition Language) interface for Extraction module.

```
struct Problem_Struct {
         string testNodeID;
         string type;
         string difficulty;
         string correctness;
};
typedef sequence<Problem_Struct> Problem_info;
interface Extraction {
         string Solving_Problem(in User_info user, in Problem_info problem);
         string Learning_Node(in User_info user, in string Next);
}
```

Figure 6. IDL interface for Extraction module

Diagnosis module diagnoses learning state using parameters from extract module. Diagnosed result is used for updating profile information. [Fig.7] is IDL interface for Diagnosis module.

```
struct Condition_Struc{
         User_info user;
         string condition;
};
typedef sequence<Condition_Struct> Condition_info;
interface Diagnosis {
         Condition_info Examine( in string User_input );
}
```

Figure 7. IDL interface for Diagnosis module

3.3 MHEG Engine on Client Side

In this section, we design MHEG engine that can support real time user interaction and standard multimedia representation. Figure 8 shows the MHEG engine configuration.

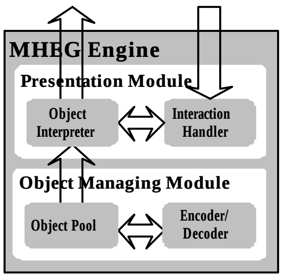

Figure 8 MHEG engine on client side

The engine consists of object managing module and presentation manager. The encoder and the decoder are based on class library. The class library serves converting between runtime objects and the MHEG objects. The presentation manager includes object interpreter and the interaction handler.

(1) Object Managing Module

In our class library, we define the base class to inherit subclass. The MH5Object class, the base class, has method that can encode and decode one BER(Basic Encoding Rule) item. Each member variable of the MH5Object class is

mapped into octets of the BER. The MH5Object class has methods for processing octets. It decodes the BER items, and provides decoded information to child classes. It also encodes information, which child classes specify, to the BER.
The root class is parent class of all other MHEG classes. The root class has an identifier of the application class. The group class includes the action object that must be executed after the MHEG application activated. Users can decode the application object by invoking member method. During encoding of the application object, we must add the BER lengths of lower level objects to length octet of higher level object. Thus, we need two encoding phases.
The scene class group defines the ingredient classes activated simultaneously. Unlike the application class, the scene class includes member variables that are required for presentation such as position and size. But, encoding and decoding of the scene object is same as them of application class.

(2) Presentation Manager

The object interpreter analyzes internal object from decoder, converts them to object lists, and then stores in the object pool. The internal object from decoder is a tree-like structure whose root is the application class. The object handler extracts behaviors of presentable objects, and relocates objects.
When the object interpreter converts objects, we use temporal and spatial information. Since objects are relocated in accordance with spatial information, spatial synchronization and simple temporal synchronization in list structure are accomplished. The object pool contains lists of internal objects.
The purpose of interaction classes, such as hypertext, entry field, button, and slider, is supporting interaction with users. The interaction handler gathers and processes user inputs and events from presentable objects. The interaction handler analyzes transmitted events, and selects action objects. After that, it analyzes those action objects, and sends appropriate messages to message handling routine. Each interaction requires an action that selects certain object or modifies one or more objects.
All input has theirs own type in accordance with the interaction classes. The interaction classes are divided into real-time interaction classes and non real-time interaction classes. Table 4 shows types of user input for each MHEG-5 user interaction class.

Table 4 Types of user input

Class	User Input	Data Type	Processing Type
Hypertext	Select/not	Boolean	real-time
Entry Field	Text String	String	non real-time
Push Button	Click/ Not	Boolean	real-time
Radio Button	Press/Release	Boolean	non real-time
Hot Spot	X, Y Position	Integer pair	real-time/ non real-time
Slider	Value	Integer	real-time

Link class consists of link condition and link effect. Link condition is a condition that invokes a link operation. It can be a combination of other conditions. Link effect is an action list that is executed when a link operates. Link effect consists of event source and event data. Event data is a user interaction class that causes an event for link operation. The interaction handler compares event data to user input. If this comparison returns true, the interaction handler executes actions specified in defined condition. Link effect is to be an elementary action or action list. Action list can include an action list. In case of this, action size, the whole size of action list, is required. Action targets, which are modified or selected by each action, are also required.

4. Conclusion and Future Work

We design multimedia system that has intelligent education capabilities on the Web. Our system has several advantages than legacy systems on the Web. We provide individual learning facilities using linguistic variables and fuzzy logic. And we provide more interactivities using MHEG standard. We can distribute educational system on the Web using CORBA. Our system is flexible for distance learning and training that requires more specific and dedicated user interaction.
Of course, We consider introduce XML to our system for multimedia representation on the Web. It accomplished, we can provide more generalized education features on the Web.

References

[Bar98] M. Barajas, A. Chrysos, A. Bosco, M. Fonollosa, I. Álvarez and M. Sancho, Sancho, "Virtual Classrooms in Traditional Universities: Changing Teaching Cultures Through Telematics," *Educational Multimedia and Hypermedia*

[Bol98] A. Bolk and D. R. Britton Jr., "The Web Is Not Yet Suitable for Learning," *IEEE Computer*, pp. 115-116. June, 1998.

[Car95] C. A. Carver Jr. and J. E. Gregory, Networked Hypermedia in an Undergraduate Curriculum, *Educational Multimedia and Hypermedia*, 1995, Association for the Advancement of Computing in Education, Charlottesville, VA. 139-144.

[Cra98] P. Cravener, "Education On the Web: A Rejoinder," *IEEE Computer*, pp. 107-108, September, 1998.

[Dwy95] D. Dwyer, K. Barbeiri and H. M. Doerr, "Creating a Virtual Classroom for Interactive Education on the Web," *International World Wide Web Conference*, 1995, Boston, MA. USA. -

[ISO95] ISO/IEC IS 13522-1 *Information technology - Coding of Multimedia and Hypermedia information - Part 1: MHEG object representation - Base notation(ASN.1)*, 1995.

[ISO96] ISO/IEC IS 13522-5 *Coding of Multimedia and Hypermedia Information - Part 5: Support for Base-Level Interactive Applications*, 1996.

[Ibr95] B. Ibrahim and S. D. Franklin, "Advanced Educational Uses of the World-Wide Web," International World Wide Web Conference, 1995, Boston, MA. USA.

[Jon90] D. H. Jonassen and H. Mandl(Eds.), *Designing Hypermedia for Learning*, Springer-Verlag.

[Kap91] F. M. Kappe, "Aspects of the Modern Multimedia Information System," Ph. D. Dissertation, Graz University of Technology, Austria.

[Mit98] S. Mittrach and G. Schlageter "A Tutoring Wizard Guiding Tutorial Work in the Virtual University," *Educational Multimedia and Hypermedia*.

[Nak95] K. Nakabayashi, Y. Koike and M. Maruyama. "An Intelligent Tutoring System on World-Wide Web : Towards an Integrated Learning Environment on a Distributed Hypermedia," *Educational Multimedia and Hypermedia*, 1995, Association for the Advancement of Computing in Education, Charlottesville, VA. 488-493.

[OMG95] OMG (1995). *CORBA 2.0 Specification*, http://www.omg.org/corba/iiop.html.

[Yoo94] K. S. Yoon and C. J. Wang, "Authoring System for the Development of ITS," IEEE TENCON, 1994, Singapore, 97-101.