ACGs: Adaptive Course Generation System - An Efficient Approach to Build E-learning Course

Anh Nguyen Viet, Dam Ho Si
Faculty of Information Technology, College of Technology
Vietnam National Uninversity Hanoi
vietanh@vnu.edu.vn , damhs@vnu.edu.vn

Abstract

Recently, adaptive course generation has been focusing by several researchers. We have built ACGs system to create adaptive courses for each learner based on evaluating demand, ability, background and learning style of them. ACGs is also separated content from model of course and extended learning objects in order to using intelligent agent to help them choice effectively adaptive course.

Keyword
ACGs, Bee - gent, E-Learning, LO.

1. Introduction

Learner in E-learning environment in order to obtain knowledge that matches with he or she demand. How learners obtain efficient knowledge is not only challenge to them but also to the tutors. To build E-learning course is following some steps which can device two phrases: to build content and structure of course. Designer advised some course models when create a course, however this model is often based on prototype so it is not suitable with many learner's demand because demand of each learner is flexible. Besides, learner has different in background, ability, and learning style.

Recently, several research is focusing on create adaptive courses. Some research obtains good result as P.Brusilovsky studies adaptive hypermedia systems. J.Vassiliva has been developing DCG system to build corporative course between tutor and learner. Y.M. Huang has been developing systems base on agent to provide resource for learner in different environment.

Goal of our ACGs is provide adaptive course for each learner. Using intelligent agent and evaluate demand, ability, background and learning style of learner to create adaptive course. In this paper we are also propose a method to extend attributes of LO as well as learner model to create adaptive course.

Next section we describes architecture of ACGs. How ACGs is implement will introduce in section 3. Section 4 is summary some results and orient research in the future.

2. Architecture of ACGs

To build adaptive course, at first, ACGs evaluate learner, through this phrase some attributes of each learner as demand, ability, background, learning style were obtain. Base on ACGs will choice content and model adapt with each learner. We design ACGs include three modules: Learner Module (LM), Content Module (CM) and View Module (VM) is described in figure 1.

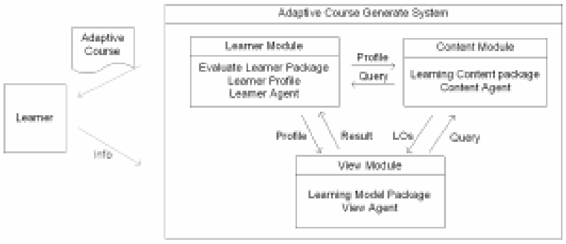


Figure 1. Architecture of ACGs

After registering a course and login successfully, ACGs uses Evaluate Learner Package (ELP) to get information from learners as well as to encourage learners to do some examination tasks. Results of evaluating learners are stored in Learner Profile (LP). Learner Agent (LA) suitable to each learner is created in order to update LP information of the learner during the course. Based on LP of each learner, the system proposes the course for the learner to choose. Selection of courses is taken by Content Agent (CA) and View Agent (VA). CA has a function to select content of course. Content of the course is stored in Learning Content Packages; CA not only selects content of the course for first time enrolled learners, during the learning process, but also cooperates with other CA in order to select contents of the course. VA has a function to select adaptive course for learner Learning Model Packages (LMP), to update new LMPs and to cooperate with other VAs.

During the course, examining and evaluating learners is implemented by ELP. This task can be carried out after each section of knowledge. Results of evaluation are updated in LP by LA. So that CA and VA will create adaptive structure of course with learner at that time. Knowledge taken in by the learner during the course can be changed frequently, so that content and model of the course also can be changed during the course.

2.1. Learner Module

Learner Module (LM) is designed in order to evaluate demands, basic knowledge, ability, skills, learning styles of leaner. LM consists of three components: Evaluate Learner Package (ELP), Learner Profile (LP), and Learner Agent (LA).

Evaluate Learner Package (ELP) has a function of managing information relating to the learner. ELP contains tests that are created in order to evaluate demands, basic knowledge, ability, skills, learning styles of learner. Information of each learner through ELP is stored in Learner Profile (LP). Each learner enrolling the course has one corresponding LP. Updating LP is carried out during learning course of the learner.

To update LP, we use Learner Agent (LA). LA has a function of managing information of the learner during the whole course and ensuring the new status of the information of the leaner.

2.1.1. Learner Profile. LP plays an important role in creating adaptive course for each learner. Each LP consists of learners attributes which is base for ACGs to create adaptive course for each learner. Attributes of LP can be classified into five categories : Identifier of Learner Profile – ILP, Course of Learner Profile – CLP, Demand of Learner Profile – DLP, Base of Learner Profile – BLP, Result of Learner – RLP. Currently, ACGs evaluate basic knowledge of learner based on the course in which the learner participates; evaluating relative courses will be studied in the near future.

Table 1. Basic attributes of LP

ILP	CLP	DLP	BLP	RLP
ID	CourseID	TimeTo Finish	LearnerCategory	Secsion ID
Name	Course Name	Strategy To Start	Pre Test	Score
Age	Date Enroll	Solving Problem		Post Test
Gender		Interactive		Threshold
Email		Focus		

When selecting attributes of LP, ACGs determines basic attributes for each learner. As selecting attributes has effects to the process of creating adaptive course for the learner, the usage of many attributes is expected to get good results, however, if using too many attributes, we may have some problems in manipulating and classifying learners in the same course.

Table 1 shows basic attributes which we use for evaluating. After the learner logins successfully, ACGs inits a unique ID in order to identify the learner. Value of attributes is basic information for classifying requirements, basic knowledge, demands.. etc of each leaner.

Table 2. Example of attributes of LP

	Attribute	Value
1	ID	*01*
2	Name	*Nguyen Viet Anh*
3	CourseID	*C01*
4	CourseName	*C++ Programing*
5	Strategy To Start	*Top –down*
…	…	...
	LearnerCategory	*Beginer*

2.1.2. Evaluate Learner Package. Updating information in LP of each learner is conducted through two information channels. The first channel obtains information from the answers of questionnaires as well as tests during the course. The second channel is automatically updated through relative LA system corresponding to each learner. We built ELP package including bank of test questions, which are classified into relative categories corresponding with DLP, BLP and RLP. For example, question for attribute LearnerCategory – classifys levels of learners: Beginner, Intermediate, Advanced and there are three options corresponding with these levels.

2.1.3. **Learner Agent.** During the course, DLP, BLP and RLP of each learner changes frequently. When the learner finished test, LA is inited and it updates information in LP coresponding with the learner until finishing the course. For example, for the first time participating the course, the leaner choose the value *Beginer of LearnerCategory*, during the course, based on the results of tests, in *t* time of the course, if the learner gets threshold δ corresponding with level *Intermediate*, LA will automatically update the value of *LearnerCategory* by the new value *Intermediate*.

2.2. Content Module

Content Module (CM) consists of two components: Learning Object Package (LOP) and Content Agent (CA).

2.2.1. Learning Object Package. LOP is designed for storing contents and models of courses. Contents of the course consist of many Learning Objects – (LO) [8, 14]. Each LO may contain one concept, one object, image, audio, etc…, or flexible combination of category files. Dublin Core Metadata Initiative [15] proposes 12 basic attributes of Learning Object Metadata (LOM).

Table 3. Basic attribute of LOM

IsPartOf	HasVersion	References	IsBasicFor
HasPart	IsFormatOf	IsReferencyBy	Requires
IsVersionOf	HasFormat	IsBaseon	IsRequiredBy

In order to select adaptive LO for learner, we supplement some attributes for LO.

Table 4. Supplementary attributes

Attribute	Attribute
Prerequisite	*DifficultyLevel*
MasterLevel	*KnowledgeExpressed*
Relation	*InteractionStyle*
RequireTime	

Content structure of the course is represented by Learning Object Graph – (LOG). The peaks of LOG can be classified into two categories: Mandatory Learning Objects (MLOs) and Secondary Learning Object (SLOs) [7]. To finish the course, learner is required to travel all peaks. SLO Peaks are prerequisite learning objects, it is not necessary for learner to travel all these peaks. Oriented vertexes represent the relation among LOS. Participating the course, learner travels LOG, usually path contains peaks of MLOs, however, there are many different paths through peaks of SLO. Selecting paths through LOS in LOGs depends on demand, ability, preference of learner, and it is also the target of ACGs system.

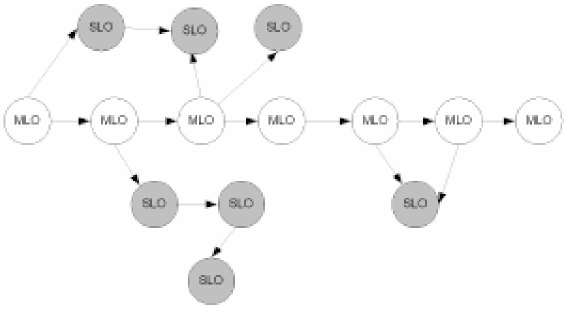


Figure 2. Structure of the content of the course

Selecting LO is a searching process of MLO, SLO peaks in LOGs which are suitable for constraints C (Requirements of learner) based on following algorithm:

```
S = []; // Set of suitable LO
For each mlo ∈ MLO
 If Constraint (mlo) then
   S= S ∪ {mlo};
   For each slo ∈ MLO
     If Constraint (slo) then
       S= S ∪ {slo};
    End
```

2.2.2. Content Agent. Content Agents (CA) have a function of communicating with LAs. Extent attributes of LO and attributes of learner are bases for LA and CA to propose suitable strategies in order to choose suitable LO for learners. Currently, ACGs do not expand interactive ability among CA corresponding to different courses in which learner participate.

2.3. View Module

View Module has a function of creating learning architecture suitable for each learner based on the combination between Learner Module and Content Module. We design VM including two components: Learning Model Package (LMP) and View Agent (VA).

2.3.1. Learning Model Package. Courses are always organized following one learning model. Winley proposed five common models: Classic tutorial, Activity –centered, Learner-customized, Knowledge – paced, exploratory tutorials [12]. ACGs classify types of learning models following these aspects: teaching styles, learning strategies, content representation.

Table 5. Classifying learning models

Aspect	Learning Model
Teaching	Help, self –study, games, tests
Learning	Top-down, bottom-up, consequence, parallel, level
Representation	Images, audio, text…

As usual, in order to finish the course, the learner must obtain fixed amount of knowledge, evaluating learner's knowledge is executed by tests. With the learning structure represented by LOG in Figure2.2, to finish this course, learner must travel all of peaks MLO. Participating the course, each learner has different requirement, so that leaning model must be different. ACGs use set of attributes (R, A, C) when builds learning model. R = [R_1, R_2,…,R_n] (Learning Object Required) is set of LO that learner must obtain when he(or she) finish the course. A (Actual Obtain Learning Object) is set of LO that the learner obtain during the course and C (Constraint of Learner) is set of requirements, constrains of learner (those requirements and constrains is represented in LP of each learner. Firstly set= [], process of building set A is conducted during the course. When finishing the course, each learner will create one learning model that is represented by objects in set

A. This learning model is supplemented into LMP package. If there are many learners participating in the course, then learning structures in LMP are more flexible. View agent updates and selects learning structure in LMP.

2.3.2. View Agent. View agent is designed in order to update, supplement learning models in LMP package, in addition, VA selects learning adaptive model for learner and communicates with others VA to build learning models. Models are built by following steps:

1 Selecting learning adaptive model in LMP for learner.
2 Selecting learning model which contains minimized LO among selected models in step 1
3 Building suitable learning model for constraints C

3. Deploying system ACGs

3.1. Tools to develop

System ACGs is developed basing on eXtensible Markup Language (XML), Bonding and Encapsulation Enhancement Agent (Bee – gent), và JAVA. ACGs uses technology XML because currently Share Content Object Reference Model (SCORM) is a standard common content in E-learning , using technology XML, and is one of our research approaches , is to expand LO attributes from available content following SCORM standard. Bee - gent [9] is one of few development application environments using pure agent technology. Bee – gent allows developer to easily build open systems based on its background, as well as to use multi autonomous agent. Bee – gent is very suitable for many agent applications, with high communication to solve problems likes ACGs. We choose JAVA technology because its is a programming language to develop Bee Agent, as well as JAVA supports many libraries to manipulate with files XML such as JAVA DOM, JAVA SAX or XSL.

3.1.1. XML. XML is a language defined by World Wide Web Consortium (www.w3c.org) [15]. XML is commonly used and became standard language for sharing information in Internet. XML designed to separate syntax and concept for easily building structure of content. XML document is organized by tree structure; each node in the tree is identified with name (element type) and set of attributes including name and value, leaves storing actual data.

Table 6. XML Data file

```
    <DOCUMENT>
        <LEARNER>
                <IID>02</IID>
                <NAME>
                        <LAST_NAME>Phung
Chi</LAST_NAME>
                        <FIRST_NAME>Dung</FIRST_NAME>

                </NAME>
                <AGE>24</AGE>
                <GENDER>Male</GENDER>

        <EMAIL>dungpc@vnu.edu.vn</EMAIL>
        </LEARNER>
    </DOCUMENT>
```

3.1.2. Bee – gent. Bee-gent system consists of 2 agents. Wrapper agent is created with application; it controls the status of applications and responds to suitable behaviors as necessary. Many Mediation agents support to negotiate ativities inside the application by message processing. Mediation agent moves from this application to other application by communicating with Wrapper agent.

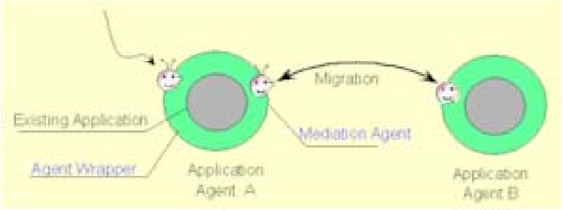


Figure 3. Action Model of Bee-gent

3.2 Steps to build ACGs system

We deploy ACGs system as follows:

1 Building ELP package for getting learners opinions.
2 Building questionares to evaluate basic knowledge of learner for each course.
3 Building LP structure for each learner.
4 Building course contents with Los having expanding extent attribute.
5 Building strategies for agents: LA, CA as VA.

In step one and two, question are designed in order to allow learners to select one or many answers options. Questions in step 1 are designed to determine values of attributes of learners in categories DLP, BLP. Questions in step 2 are designed to determine value of attributes in category RLP. Expanding attributes of LO in 4 steps base on attributes in table 4. As follows, we show step 3 and step 5 in process of creating dynamic learning course.

3.2.1. Representing Learner Profile by XML. Learner's attributes in table 1 are represented by formatting file XML. For each participating learner, ACGs provide one suitable profile until finishing the course.

Table 7. Learner profile

```
<!ELEMENT DOCUMENT (LEARNER_PROFILE)>
<!ELEMENT                   LEARNER_PROFILE
(ILP,CLP,DLP,BLP,RLP)>
<!ELEMENT ILP (IDD,NAME,AGE,GENDER,EMAIL)>
```

```
<!ELEMENT                                    CLP
(COURSE_ID,COURSE_NAME,DATE_ENROLL)>
<!ELEMENT                                    DLP
(TIME_TO_FINISH,STRATEGY_TO_START,
SOLVING_PROBLEM,     INTERACTIVE,     FOCUS,
FLEXIBILITY)>
<!ELEMENT BLP (LEARNER_CATEGORY,PRE_TEST)>
<!ELEMENT                                    RLP
(SECTION_ID,SCORE,POSTEST,THRESHOLD>
<!ELEMENT IDD (#PCDATA)>
<!ELEMENT NAME (FIRST_NAME,LAST_NAME)>
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>
<!ELEMENT AGE (#PCDATA)>
<!ELEMENT GENDER (MALE|EMALE)>
<!ELEMENT MALE (#PCDATA)>
<!ELEMENT FEMALE (#PCDATA)>
<!ELEMENT EMAIL (#PCDATA)>
<!ELEMENT COURSE_ID (#PCDATA)>
<!ELEMENT COURSE_NAME (#PCDATA)>
<!ELEMENT DATE_ENROLL (#PCDATA)>
<!ELEMENT TIME_TO_FINISH (#PCDATA)>
<!ELEMENT                    TRATEGY_TO_START
(IN_ORDER|O_ORDER>
<!ELEMENT IN_ORDER (#PCDATA)>
<!ELEMENT NO_ORDER (#PCDATA)>
<!ELEMENT                     SOLVING_PROBLEM
(TOP_DOWN|BOTTOM_UP|IERARCHY|LAT)>
<!ELEMENT TOP_DOWN (#PCDATA)>
<!ELEMENT BOTTOM_UP (#PCDATA)>
<!ELEMENT HIERARCHY (#PCDATA)>
<!ELEMENT FLAT (#PCDATA)>
<!ELEMENT                         INTERACTIVE
(COLLABORATITVE|INDIVIDUAL)>
<!ELEMENT FOCUS (PARALLEL|SEQUENTIAL)>
<!ELEMENT                         FLEXIBILITY
(LIBERAL|CONSERVATIVE)>
<!ELEMENT                    LEARNER_CATEGORY
(BEGINER|INTERMEDIATE|ADVANCED)>
<!ELEMENT PRETEST (#PCDATA)>
<!ELEMENT SECTION_ID (#PCDATA)>
<!ELEMENT SCORE (#PCDATA)>
<!ELEMENT POSTEST (#PCDATA)>
<!ELEMENT THRESHOLD (#PCDATA)>
```

3.2.2. Action model of agents. Agent in ACGs system communicates with each other following model showed in Figure 4 by three stages:

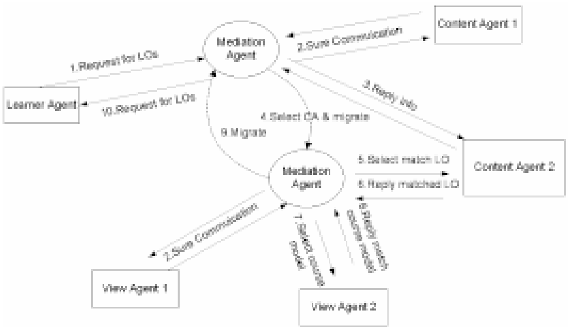
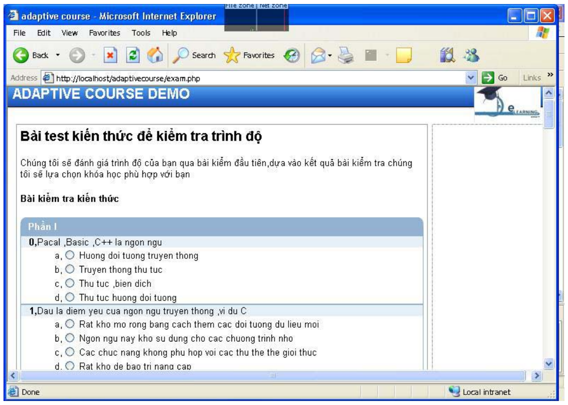


Figure 4. Activity agent model

1. LA relevant to each learner requires choosing LO.
2. Mediation agent is created in order to communicate with CAs.
3. Feedback information between CA and Mediation agent ensures transferal good information interaction.
4. Mediation agent plays a role as agent wrapper negotiates
5. Selecting LOs.
6. Results of selecting process in step 5 are LOs which meet the conditions.
7. Selecting learning structures.
8. Results of selecting process in step 7 are suitable learning structure with learner.
9. Mediation agent plays a role as wrapper agent sends results to LA.
10. The course meets the demands of learner.

4. Our experiments and Results

We built ACGs base on web application. We created two courses: C++ programming and Data structure & Algorithm. Each course has fifteen sections. When learner participates in the course, he or she was given a questionnaire and pre – test.


adaptive course - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Done
Local intranet

http://localhost/adaptivecourse/exam.php

ADAPTIVE COURSE DEMO

Bài test kiến thức để kiểm tra trình độ

Chúng tôi sẽ đánh giá trình độ của bạn qua bài kiểm đầu tiên,dựa vào kết quả bài kiểm tra chúng tôi sẽ lựa chọn khóa học phù hợp với bạn

Bài kiểm tra kiến thức

Phần I

0,Pacal ,Basic ,C++ la ngon ngu

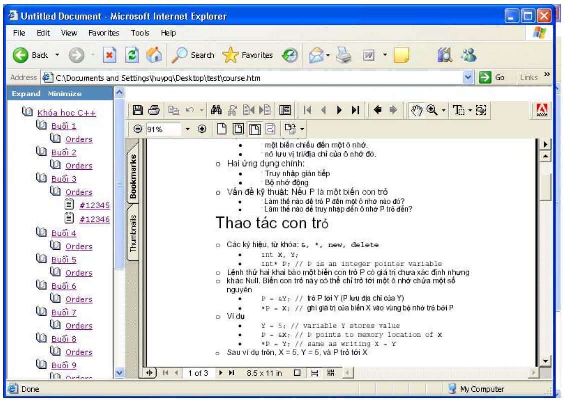
a, ○ Huong doi tuong truyen thong
b, ○ Truyen thong thu tuc
c, ○ Thu tuc ,bien dich
d, ○ Thu tuc huong doi tuong

1,Dau la diem yeu cua ngon ngu truyen thong ,vi du C

a, ○ Rat kho mo rong bang cach them cac doi tuong du lieu moi
b, ○ Ngon ngu nay kho su dung cho cac chuong trinh nho
c, ○ Cac chuc nang khong phu hop voi cac thu the the gioi thuc
d, ○ Rat kho de bao tri nang cap

Figure 5. The test for learner

The ACGs had generated the course that appropriate for learner. In the course content, there is a test in each section. The following contents will be adapting base on the test results.



C:\Documents and Settings\huypq\Desktop\test\course.htm

Khóa hoc C++
Buổi 1
Orders
Buổi 2
Orders
Buổi 3
Orders
#12345
#12346
Buổi 4
Orders
Buổi 5
Orders
Buổi 6
Orders
Buổi 7
Orders
Buổi 8
Orders
Buổi 9
Orders

- một biến chiều đến một ô nhớ.
- nó lưu vị trí/địa chỉ của ô nhớ đó.

○ Hai ứng dụng chính:

- Truy nhập gián tiếp
- Bộ nhớ động

○ Vấn đề kỹ thuật: Nếu P là một biến con trỏ

- Làm thế nào để trỏ P đến một ô nhớ nào đó?
- Làm thế nào để truy nhập đến ô nhớ P trỏ đến?

Thao tác con trỏ

○ Các ký hiệu, từ khóa: `&, *, new, delete`

- `int X, Y;`
- `int* P; // P is an integer pointer variable`

○ Lệnh thứ hai khai báo một biến con trỏ P có giá trị chưa xác định nhưng
○ khác Null. Biến con trỏ này có thể chỉ trỏ tới một ô nhớ chứa một số nguyên

- `P = &Y; //` trỏ P tới Y (P lưu địa chỉ của Y)
- `*P = X; //` ghi giá trị của biến X vào vùng bộ nhớ trỏ bởi P

○ Ví dụ

- `Y = 5; // variable Y stores value`
- `P = &X; // P points to memory location of X`
- `*P = Y; // same as writing X = Y`

○ Sau ví dụ trên, X = 5, Y = 5, và P trỏ tới X

Figure 6. The C++ programing course

There are approximately seventy student partacipating the course. After finished the course, we survey for all student who had partacipated the course. The survey evaluate ACGs in Struture, Interface, Adaptive ability, and Meet learner demand ability.

Table 6. The results of survey

	Excellent	Good	Fair
Structural	70%	20%	10%
Interface	35%	50%	25%
Adaptive	85%	5%	10%
Meet Demand	65%	20%	15%

5. Conclusion

This paper evaluated learners base on some attributes, as well as expands some attributes for learning object LO and separates contents from learning model when building course. Base on that, building ACGs system creates adaptive courses for different learners through intelligent agents. ACGs system gets the following results:

a. Flexibility in classifying learning through attributes in LP, easily expanding attributes for evaluating learner by XML.

b. Reusing content packages following SCORM standard.

c. Huge database of learning model is helpful for learner to select during updating new learning models when he or she finishes the course.

In the future researches, we will focus on evaluating learner in corporative learning among learners in network environment.

6. References

[1]. Hao-Chuan-Wang and Tsai -Yen Li, "Considering Model - based adaptively for Learning Objects"*, Learning Technology Vol.6*, 2004, *9-12.*

[2]. Y.M Huang and J.N Chen and K.T Wang and C.H Fu, "Agent - Based Web Learning System", *Learning Technology Vol.6,* 2004, 38- 42.

[3]. G.McCalla and J.Vassileva and J.Greer and S.Bull, "Active Learning Modeling", *ITS,* 2000.

[4]. J.Vassileva, "Dynamic Course Generation on WWW", *AIED,* 2000.

[5]. P.Brusilovsky, "The Construction and Application of Student Model in Intelligent Tutoring System", *Journal of Computer and System Sciences International 31(1)*, 1994

[6]. B.A Charles and A. David, "Multimedia Intelligent System for Online Learning", *Proceedings of the International Conference on Computers in Education*, 2002.

[7]. Y. Atif and R. Benlamri and J.Berri, "Dynamic Learner Modeler", *Educational Technology & Society 6(4)*, 2003, 60-72.

[8]. M. D Merrill, "Knowledge Objects", *CBT Solution*, 1998, 1-11.

[9]. Bee – gent, software at http://www2.toshiba.co.jp/beegent/index.htm

[10]. M. Altenhofen and J. Schaper, "Flexible instructional strategies for E- learning", *Proceedings of 35th Hawaii International Conference on System Sciences*, 2002.

[11]. S.C. Buraga, "Developing agent- oriented e-learning systems", *AIED*, 2003.

[12]. William Horton, *Designing Web-Based Training*, Published by Willey Inc, 2000.

[13]. W3C, Extensible Markup Language, software at www.3c.org

[14]. P. Pickard and K.Fisher and R.Jones, "Learning Object for Introductory Computer Programming", *Learning Technology Vol.6*, 2004

[15]. G. Knolmayer, "Decision Support Models for Composing and Navigating through e-Learning Objects", *Proceedings of the 36th Hawaii International Conference on SystemEducation*, 2003.