Proceedings of the 5th WSEAS Int. Conf. on DISTANCE LEARNING AND WEB ENGINEERING, Corfu, Greece, August 23-25, 2005 (pp158-163)

A Course Blueprint Designs using Learning Objects Approach and XML-Based Planning for Multi-Instructors Learning Systems

NOPPAMAS PUKKHEM[1] AND WIWAT VATANAWOOD[2]
Department of Computer Engineering, Faculty of Engineering
Chulalongkorn University
THAILAND

Abstract: - This paper proposes a course blueprint construction model, by using XML-based plan that represents knowledge of ordering rule of learning objects composition. Hierarchical structure is used as the structure for the composition of learning objects in each composite level. Based on Goldsmith's closeness index, we implement the ordering algorithm to find the integration of different course orders. There are four objectives in this work. First, to propose the procedure of analysis the course blueprint design. Second, to presents the two ways of ordering method; front set method and rear set method, which are used to evaluate and establish course object order. Third, to illustrate the method of integrated various orders from different designs. Finally, to present the model for course blueprint construction with component-based development concept based on model-view-controller (MVC) architecture and supported technologies. This approach reduces cost and times for constructing course instances from efficiency blueprint and also solves the different ordering problems among instructors.

Key-Words: - Course Blueprint, Learning Objects Composition, Multi-Instructors Learning System, Ordering Method, XML-Based Plan

1 Introduction

Adaptive learning is one of the primary goals of e-learning. The dynamic just-in-time generation of learning resources based on the beliefs, desires, and goal of a student to facilitate learning [1]. Web-based distanced education systems are complex. As mentioned by many authors, these systems integrate many interrelated processes, aggregate a large set of components, and can be considered from a wide range of viewpoints, paradigms and disciplines [4]. The granularity of an atomic object is entirely up to its author. An atomic object can be a piece of text, such as a whole book, a chapter or a section of a book, a paragraph or even a phase; similarly, it can be an image or part of an image, and so on [9].

Each course in curriculums in higher education is delivered at a fixed duration of time and in a quite steady learning path. However, different characteristic of instructional designer and learner while design the same course may vary in instructional paths, contents and sequence [5]. These variations may result in hard to meet to different requirements. We use the concept of blueprint design in civil engineering to apply for construct the suitable location of course objects, called course blueprint. As we mentioned earlier, atomic learning objects are located in the provided building blocks called composite course objects. These course objects used to be assigned in course blueprint must have characteristics derived from the IEEE Learning Object Metadata (LOM) standard. As the result, this model provides a reusable course blueprint for various characteristics of instructors and learners. If the instructional designer requirements do not meet the expectations, the course blueprint is re-planned.

There are two cases to be considered when composing course object: to replace a course object in an existing workflow with a similar functionality and to define a whole new workflow using available relevant course objects in course object repository. In the following, we choose the first case to demonstrate our model. In addition, based on Goldsmith's closeness index, we also propose an ordering algorithm that is used to integrate different orders.

The remainder of this paper is organized as follows. In section 2, we briefly mention the importance of Learning Object in E-learning systems. Section 3 shows the course blueprint design of this work. Section 4 shows the proposed of course blueprint construction model. Section 5 gives the conclusion of this work and finally, section 6 suggests for the future research.

2 Learning Objects

E-learning system brings some new characteristics regarding to classical training situations. In distance education situations, the learners have to be more active than in classical training applications. Knowledge is not presented linearly as in a course [2]. It is divided into Learning Objects and the users have to choose the Learning Objects they consider as being useful for them.

A Learning Object is a self-contained or self-descriptive learning resource that is appropriately tagged according to the standards of metadata [7]. It is located with the aid of standards compliant metadata indexing and searching services [10].

A Learning Object currently leads other candidates for the position of technology of choice in the next generation of instructional design, development, and delivery, due to its potential for reusability, generatively, adaptability, and scalability [12].

3 Course Blueprint Design

In automatic course blueprint construction from planning knowledge, the main idea is to generate a layout of learning object suited to the needs of different instructors. If the instructional designer requirements do not meet the expectations, the course blueprint is dynamically re-planned.

3.1 Course Structure

We use the hierarchical structure that enables flow of control horizontally and left to right to present the view of course blueprints. In this work we call the learning objects as course objects and divide them into four groups: course, part, chapter, and module. We can map our content model on IEEE LOM content model (raw media, learning object, complex learning object) and presents the mapping course object in each group of this work to learning objects aggregation level according to IEEE LOM standard [6] in table 1.

3.2 The Integrated Difference Course Object Ordering Mechanism

Table 2 presents the several modules contained in one chapter and defined code from instructor. As vary ordering from different instructors shown in fig.1, we need the mechanism that used to integrate the different course object ordering.

Table 1. Mapping course object to learning objects aggregation level according to IEEE LOM standard.

IEEE LOM Element	Value Space	Description	Course Object
General/ Aggregation Level	1	The smallest level of aggregation, e.g. raw media data or fragment	Module
	2	A collection of level 1 learning objects, e.g. a lesson chapter or a full lesson	Part, Chapter
	3	A collection of level 2 learning objects, e.g. a course.	Course

Table 2. Example modules contained in chapter[11].

Chapter 8 (Ch8): Building the analysis model	
Module name	**Code**
Scenario-Based Modeling	A
Analysis Modeling Approaches	B
Data Modeling Concepts	C
Flow-Oriented Modeling	D
Class-based Modeling	E
Create a behavioral Modeling	F

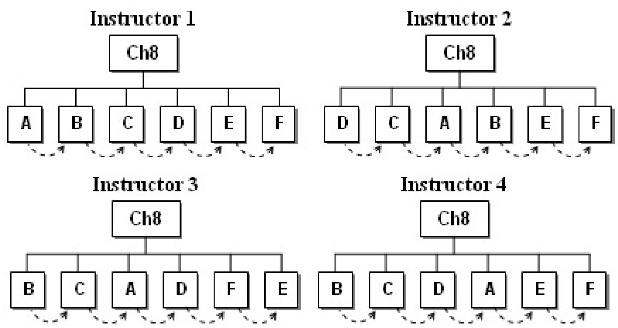


Fig.1. Different course objects ordering of four instructors in module level

Next, we present the calculation of closeness index and confidence value, and two types of concept ordering processes. They are ordered based on expectation at front and rear located confidence. We use the same algorithm to order the course object in each level.

3.2.1 Calculation of Closeness Index and Confidence Value

This way we show the algorithm to calculate the closeness index and confidence value. The procedure is discussed by an example (INS3 and INS4) and shown as follows.

Table 3. Comparison results between two instructors (INS3 and INS4)

Module	INS3		INS4		UF_{34}	UR_{34}	IF_{34}	IR_{34}	CF_{34}	CR_{34}
	F set	R set	F set	R set						
A	{B,C}	{D,E,F}	{B,C,D}	{E,F}	{B,C,D}	{D,E,F}	{B,C}	{E,F}	0.667	0.667
B	∅	{C,A,D,F,E}	∅	{C,A,D,E,F}	∅	{C,A,D,E,F}	∅	{C,A,D,E,F}	1.000	1.000
C	{B}	{A,D,F,E}	{B}	{D,A,F,E}	{B}	{A,D,F,E}	{B}	{A,D,F,E}	1.000	1.000
D	{B,C,A}	{F,E}	{B,C}	{A,F,E}	{B,C,A}	{A,F,E}	{B,C}	{F,E}	0.667	0.667
E	{B,C,A,D,F}	∅	{B,C,A,D}	{F}	{B,C,A,D,F}	{F}	{B,C,A,D}	∅	0.800	0
F	{B,C,A,D}	{E}	{B,C,D,A,E}	∅	{A,B,C,D,E}	{E}	{B,C,A,D}	∅	0.800	0
Total									4.934	3.334

Process 1: Closeness index calculation

The Goldsmith's method is used to calculate the closeness index and ordering processes [5]. We repeat to process this method in every pair of instructors. The methods are explained as follows.

- Determine all course objects N.
- For each course object (CO) belonging to N
  - Consider the ordering from INS_j and INS_k
  - For every target course objects of each instructor, insert every front course objects to front set (F set) and every rear course objects to rear set (R set).
  - Calculate the union front and rear set (UF_{jk} and UR_{jk}), and intersection front and rear set (IF_{jk} and IR_{jk}).
  - Calculate the closeness coefficient (CF_{jk}, CR_{jk}) with equation(1).

$$CF_{jk} = IF_{jk}/UF_{jk} \ \text{ or } \ CR_{jk}=IRk_{jk}/UR_{jk} \qquad (1)$$

where

UF_{jk} is number of members in union front set
UR_{jk} is number of members in union rear set
IF_{jk} is number of members in intersection front set
IR_{jk} is number of members in intersection rear set

For example, we calculate the closeness coefficient into two set types: front and rear, with equation (1). The comparison between instructor INS_3 and INS_4 is derived from the example in fig.1. The front set closeness coefficient (CF_{34}) and rear set closeness coefficient (CR_{34}) is shown in table 3.

- Calculate the closeness index between instructor number j and instructor number k, $C(INS_j,INS_k)$, with equation(2) for front set method and equation(3) for rear set method (eg. $C(INS_{3j},INS_4)$ = (1/6)*(4.934) = 0.8222).

$$C(INS_j,INS_k) = (1/N)\Sigma CF_{jk} \qquad (2)$$

$$C(INS_j,INS_k) = (1/N)\Sigma CR_{jk} \qquad (3)$$

where N is number of all nodes

Process 2: Confidence value calculation

We compute the confidence values of instructors by using closeness index from table 3 with equation (2) or (3). The results are shown in table 4; (a): front set confidence values and table 4(b): rear set confidence value.

$$Total(INS_j) = \sum_{k=1}^{m} C(INS_j, INS_k) \qquad (4)$$

$$Confidence(INS_j) = \frac{Total(INS_j)}{\sum_{i=1}^{m} Total(INS_i)} \qquad (5)$$

where m is number of all instructors

Table 4. Results of confidence calculated.

INS#	INS1	INS2	INS3	INS4
INS1		0.3889	0.5167	0.5278
INS2	0.3889		0.3222	0.3889
INS3	0.5167	0.3222		0.8222
INS4	0.5278	0.3889	0.8222	
Total	1.4334	1.1000	1.6611	1.7390
Front set Conf.	0.2416	0.1854	0.2800	0.2931

(a) Front Set confidence values

INS#	INS1	INS2	INS3	INS4
INS1		0.6500	0.5250	0.7695
INS2	0.6500		0.2833	0.7111
INS3	0.5250	0.2833		0.5556
INS4	0.7695	0.7111	0.5556	
Total	1.9445	1.6445	1.3639	2.0361
Rear set Conf.	0.2782	0.2353	0.1952	0.2913

(b) Rear Set confidence values

3.2.2 Integration of course object order

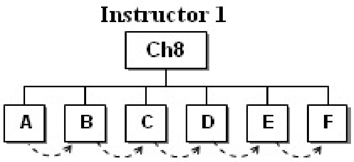
The steps of the integrated algorithm are explained as follow.

Step 1: Different instructor define the priority value of each module. We proposed two type of this step.

If we focus at front set type, the equation (6) is used to calculate the priority value. In contrast, we use equation (7) to support rear set type. Example of this step shows in fig.2(a) and fig.2(b).

$$\textit{Priority value of position}(i) = N-(i-1) \qquad (6)$$

$$\textit{Priority value of position}(i) = i \qquad (7)$$



Module	A	B	C	D	E	F
Priority	6	5	4	3	2	1

(a) Front Set Defined

Module	A	B	C	D	E	F
Priority	1	2	3	4	5	6

(b) Rear Set Defined

Fig.2. Priority values define by ordering.

As the results that mention above, equation (8) and (9) are used to compute the ordering value(OV). We define the total ordering value of each module as course object ordering value to this target module. Table 5 and table 6 offer the results of ordering value of each module.

$$OV_{nj} = Confidence(j) \times Priority(n) \qquad (8)$$

$$Total\ OV_{nj} = \sum OV_{nj} \qquad (9)$$

where n = *Target node*, j = *Instructor j*

Step 2: We represent the integrated order using the ordering value as hierarchical structure. Fig.3(a) show the result from front set method and fig.3(b) shows the result from rear set method.

As we propose the two mechanisms that provide the same ordering results, the author can implement the XML-based plan based on the selected one.

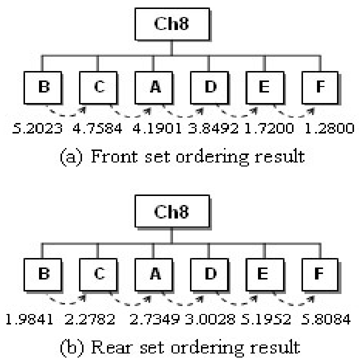


Fig.3. Hierarchical ordering of modules

The ordering processes are running in every levels of the course structure. In fig.4, ordering process is started at module ordering level and repeating in each higher level.

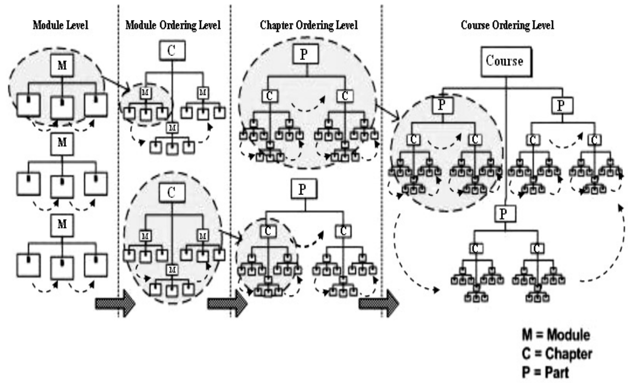


Fig.4. Overview of completely ordering process

3.3 Course Blueprint Structure

A course blueprint structure is divided into four levels (course, part, chapter, and module). Content in lower level can be aggregated as a course objects in higher level.

In our work, we define the available locations to paste any course object as three levels: part location, chapter location, and modules location. An example of diagram of module course objects is shown in fig.5.

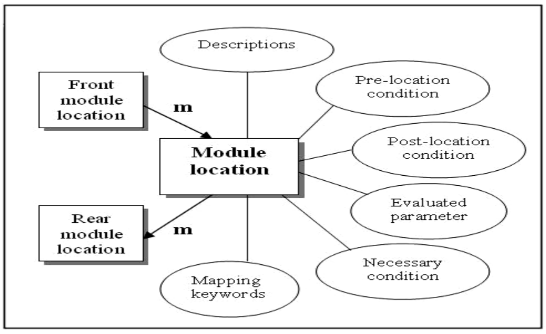


Fig.5. An example diagram of module course object location and its attributes

Front location: The location that locates before target location in same level.
Rear location: The location that locates after target location in same level.
Description: The details that describe the characteristic of course object at target location.
Evaluated parameter: The parameter that define the difficult level of content.
Pre-location condition: The parent location that locates upper one level from target location.
Post-location condition: The child locations that locate lower one level from target location.
Mapping keywords: Keywords are used to support the course objects searching from the repository.

Table 5. The ordering value using front set method.

	Confidence	Priority A	OV_A	Priority B	OV_B	Priority C	OV_C	Priority D	OV_D	Priority E	OV_E	Priority F	OV_F
INS1	0.2416	6.0	1.4495	5.0	1.2079	4.0	0.9663	3.0	0.7247	2.0	0.4832	1.0	0.2416
INS2	0.1854	4.0	0.7415	3.0	0.5562	5.0	0.9269	6.0	1.1123	2.0	0.3708	1.0	0.1854
INS3	0.2800	4.0	1.1198	6.0	1.6798	5.0	1.3998	3.0	0.8399	1.0	0.2800	2.0	0.5599
INS4	0.2931	3.0	0.8792	6.0	1.7585	5.0	1.4654	4.0	1.1723	2.0	0.5862	1.0	0.2931
Total			**4.1901**		**5.2023**		**4.7584**		**3.8492**		**1.7200**		**1.2800**

Table 6. The ordering value using rear set method.

	Confidence	Priority A	OV_A	Priority B	OV_B	Priority C	OV_C	Priority D	OV_D	Priority E	OV_E	Priority F	OV_F
INS1	0.2782	1.0	0.2782	2.0	0.5564	3.0	0.8347	4.0	1.1129	5.0	1.3911	6.0	1.6693
INS2	0.2353	3.0	0.7059	4.0	0.9412	2.0	0.4706	1.0	0.2353	5.0	1.1765	6.0	1.4118
INS3	0.1952	3.0	0.5855	1.0	0.1952	2.0	0.3903	4.0	0.7806	6.0	1.1709	5.0	0.9758
INS4	0.2913	4.0	1.1653	1.0	0.2913	2.0	0.5827	3.0	0.8740	5.0	1.4567	6.0	1.7480
Total			**2.7349**		**1.9841**		**2.2782**		**3.0028**		**5.1952**		**5.8048**

Necessary condition: The condition that defines the necessary of target location. There are two choices: <option>, which means may skip in course or <mandatory>, which means must appear in course.

To consider the necessary condition, we use the equation (10) and (11) to calculate the necessary weight. The choice that has more weight is selected to define any necessary condition in XML-based plan.

$$CO_{Total}(man) = \frac{\sum_{k=1}^{m}\left(CO_k(man) * w_k\right)}{\sum_{k=1}^{n} w_k} \quad \textbf{(10)}$$

$$CO_{total}(opt) = \frac{\sum_{k=1}^{m}\left(CO_k(opt) * w_k\right)}{\sum_{k=1}^{m} w_k} \quad \textbf{(11)}$$

where

w_k *is the weighted value for instructor k.*

$CO_k(man)$ *is the necessary of course object location (assign by Instructor k). If <mandatory>=true then* $CO_k(man)$ *= 1 else* $CO_k(man)$ *=0.*

$CO_k(opt)$ *is the optional of course object location (assign by instructor k). If <option>=true then* $CO_k(opt)$ *=1 else* $CO_k(opt)$ *=0.*

The useful of necessary condition appears when execute time. The dynamic selection course object process can skip to paste the course object that has <option> necessary condition. As this result, we can generate the different course instance supporting the various characteristic of learner.

4 Proposed Course Blueprint Construction Model

This section describes what an integrated model of hierarchical structure that is created by our ordering mechanism and rule-based planning will look like. Finally, we present how it can be done under our assumptions.

4.1 Implementing Course Blueprint Plan with XML

Attributes of location are used to create the plan of the course object composition for each level. The XML format is chosen to describe the flow of our plan. The part of course blueprint schema that used to generate the plan in shown in fig. 6.

The XML-based plan describe the characteristics of the target content, LOlink told us about the location that collect them, and provided ordering plan for composite course objects in each level.

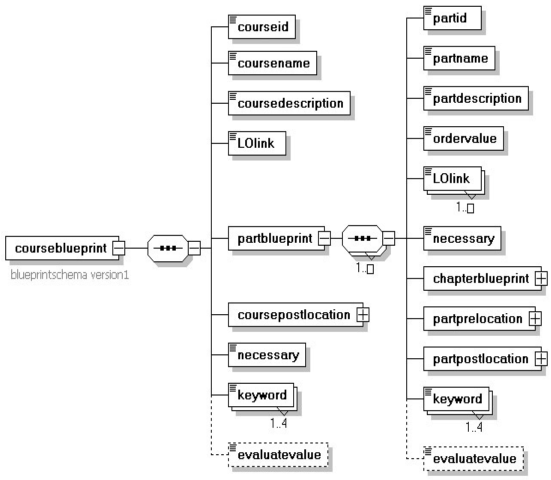


Fig.6. The part of course blueprint schema

4.3 Course Blueprint Construction Model

We design a support environment system for creating course blueprint. The system helps authors create sharable and reusable course blueprint.

We used apache xindice XML database to collect all course blueprints, because it support XML file collection as hierarchical view.

The prototype is experimentally implemented based on Model-View-Controller model using supported technologies.

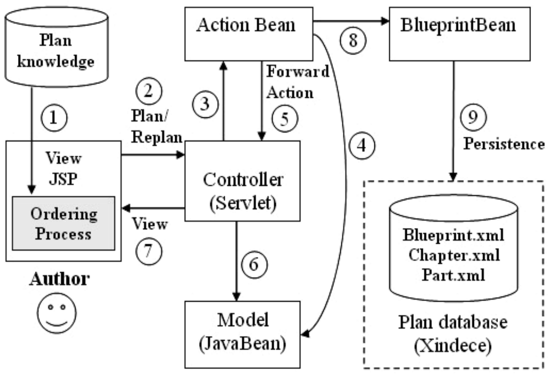


Fig.7. Course blueprint construction model

Our experiment is shown in fig.7. All processes work step by step follow as the ordering of numbers (process number 1 - 9). The three main parts in this model are controller(servlet), view (JSP) and model (javabean). The result of this system is the persistence status of course blueprint. All of course blueprint bean status, are automatically translated into XML-based plan and finally, the course blueprint is stored in plan database. It also provides reusability and adaptability in the future.

5 Conclusion

We propose a model for e-Leaning course blueprint construction. The blueprint consists of course object locations within four levels, which are module, chapter, part, and course.

Concerning the integrated learning object approach, we focus on the creating of the plan called "Blueprint plan", to guide the best suitable order of the course objects. The workflow of composition and the ordering algorithms are discussed to solve the different ordering problems among instructors.

To demonstrate course objects locations as course blueprint, we use hierarchical structures to show the level of course objects composition. The experiment is conducted by implementing a prototype for testing this model based on MVC architecture.

6 Future works

We plan to focus how agents can be used to work with AI planning and how they improve our current mechanism. We aim to implement the "adaptive course", where any path among courses can be better chosen by examining the characteristics of different learners and the requirements of instructional designers.

References:

[1] ASTD & SmartForce, A field guide to Learning Object,2002. Http://www.learningcircuits.org/ 2002/jul2002/smart-force.pdf.

[2] B.W. Ruttenbur, G.C. Spickler and S. Luric. Elearning – The Engine of the knowledge Economy, *eLearning Industry Report*, Jul/2000.

[3] C.J. Hamel and D. Ryan-Jones, Designing Instruction with Learning Objects, *International Journal of Educational Technology*, 3(1), Nov, 2002.

[4] G. Paquette and I. Rosca, Organic Aggregation of Knowledge object in Educational Systems, *Canadian Journal of Learning and Technology*, Volume 28(3) Fall/automne 2002.

[5] H.Y. Shyu, S. H. Hsich and Y. H. Chon, Integrating concept map into designing a courseware management system, Paper presented to ED-MEDIA 2003.

[6] IEEE. IEEE Standard for Learning Object Metadata, 1484.12.1-2002.

[7] LTSC, Draft Standard for Learning Object Metadata, IEEE 1484.12.1-2002, 2002.

[8] L. Baruque, R. Melo, Learning Theory and instructional design using learning object, The ED-MEDIA conference, Hawaii, 2003.

[9] P. Rigaux and N. Spyratos, SeLeNe Report: Metadata Management and Learning Object Composition in self eLearning Network, Http://www.dcs.bbk.ac.uk/selence/reports.

[10] R. Koper, Modeling units of study from a pedagogical metamodel behind EML, 2001. Http://eml.ou.nl/introduction/articles.htm.

[11] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 6th Ed, McGraw-Hill, 2005.

[12] T.Kilby, Learning Objects, Web Based Training Information Center,2002. Http://www.webbased Training.cm/trend_objects.asp.