

# The Concept Path Combination Model for Supporting a Personalized Learning Path in Adaptive Educational Systems

Noppamas Pukkhem<sup>1</sup>, Martha W. Evens<sup>2</sup> and Wiwat Vatanawood<sup>3</sup>

*Department of Computer Engineering, Chulalongkorn University, Thailand<sup>1,3</sup>*

*Department of Computer Science, Illinois Institute of Technology, USA<sup>2</sup>*

*Email: noppamas.p@student.chula.ac.th<sup>1</sup>, evens@iit.edu<sup>2</sup>, wiwat@chula.ac.th<sup>3</sup>*

## Abstract

*Personalized learning paths are becoming more common in adaptive learning systems. In this paper we extend our earlier work by developing the Content Path Combination Model (CPCM) for integrating the design paths obtained from different instructional designers and making a strongly recommended course object sequence; presenting methodologies for handling two central tasks: (i) creating the linear sequence extensions that are the integrated topological sorts of different concept path graphs; (ii) creating an integrated concept path using CPCM. The demonstrated results show that the personalized learning path is strongly supported by the synthesized path based on the work of various instructional designers and on the learner knowledge space.*

**Keywords:** Concept Path Graph, the Concept Path Combination Model, Linear Sequence Extension, Personalized Learning Path.

## 1. Introduction

The development of adaptive learning environments has become a very important area of educational technology. These environments produce educational resources that are practically unlimited, since they enable a methodology for creating personalized learning systems, distance learning systems, and collaborative work systems [4]. Web-based courseware typically creates problems in locating, selecting, and identifying semantic relationships between suitable learning resources. Web-based courses have high drop-out rates because the numbers of those dissatisfied with courses that do not engage the learners are on the rise [3].

In recent work, we have constructed an authoring tool called the “Web-Based Course Component Authoring Tool: W2CAT” to develop content objects

or course objects based on a component-based development approach [8]. W2CAT provides course objects that are typically assigned in a course blueprint, which is an instructional plan to guide the sequencing of course objects. The course blueprint describes the course object sequence that is generated by the sequence pattern combination model (SPCM), which can be used to integrate a variety of design patterns from different instructors [9]. We call the result of this process the “integrated master pattern.” In addition, these course objects must be defined in terms of the IEEE learning object metadata (LOM) standard [2] and they can also be mapped to sharable content objects (SCOs) in the SCORM content aggregation model.

In this paper, we address the way in which instructional designers use the integrated master pattern to support personalized learning paths by using the concept path combination model (CPCM). We have two central tasks: First, we want to propose a model that can combine the instructional designer recommendations that are inherited from the learning goal hierarchy with the learner’s preferences using knowledge to provide the best integrated learning path for each learner. Second, we also want to demonstrate how our method can be used for identifying the personalized learning path in adaptive educational systems.

The paper is structured as follows: In Section 2, we describe the development of an instructional strategy, with special emphasis on sequencing the course objects. In Section 3, we give an overview of the adaptive educational hypermedia system. In Section 4, we explain the Concept Path Combination Model (CPCM) analysis and design. In Section 5, we present a way to integrate the Concept Path Combination Model with the personalized learning path model. Section 6 gives our conclusions and our plans for future work.

## 2. Developing an Instructional Strategy

An Instructional Strategy addresses the way in which a course designer identifies how information will be shown to the learners and how instruction will engage the learners [15]. The main processes in the development of an instructional strategy cover various aspects: designing the sequencing and grouping clusters of content, describing learning materials that will be combined in the instructional process, establishing the lesson structure, selecting learning objects for delivering instruction, finding suitable content for the learners and specifying how learners will satisfy their learning goals.

An instructional strategy is used by many developers in supporting many tasks in an adaptive educational hypermedia system. For example, the Personal Reader system uses an instructional strategy to construct a personalized learning object reader to support personalized e-learning on the Semantic Web. It is based on configurable and reusable personalization services [14].

## 3. Adaptive Educational Hypermedia Systems

The adaptive educational hypermedia system is a model for hypermedia based systems that was designed for general purpose adaptive web applications.

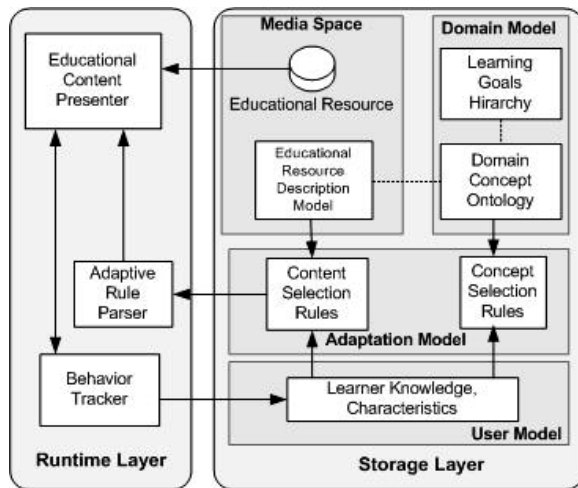


Figure 1. The generalized architecture of adaptive educational hypermedia systems [12]

For example, adaptive education hypermedia systems such as Interbook[11], NetCoach, AHA! [12], WINDS[5], Personal Reader [14] and KBS Hyperbook [6], require every hypermedia page to be connected to the domain model concept in order for the server to know when to present them in an adaptive manner [11]. Figure 1 shows the generalized architecture of adaptive education hypermedia systems which involves four key steps [12].

1. Designing the Domain Model; this process produces a design of the hierarchical learning goals, describing the course domain concepts.
2. Designing the User Model; this process produces the design of the model that defines the learner's cognitive characteristics and preferences.
3. Designing the Media Space; the process produces the design of the resource description model for representing the educational characteristics of the learning resources, e.g., the learning resource type, its complexity, the relationships among learning resources.
4. Designing the Adaptation Model; the process produces the design of the concept selection rules that are used for selecting appropriate resources from the Domain Model and the Media Space.

## 4. Our Model Analysis and Design

In our work, the concept graph is the domain model that represents all possible sequences of concept learning defined by all of the instructional designers. The domain model stores knowledge about the course preferences and characteristics. We designed the Concept Path Combination Model (CPCM) to handle differences in the way instructional designers design the learning path graph.

### 4.1. Defining Terminology

The terms used in this paper are explained as follows:

**Concept:** In this work, we define a concept as an item of knowledge for describing the subject domain and we create a node to represent each concept.

**Concept Path Graph:** A concept path graph is a directed acyclic graph that represents the set of sequencing rules that determine the order of the concepts; they should be followed by a list of the behaviors that the instructional designer intends for the learner to acquire. The learning goals are used to define the pattern of the concept path based on the domain model and user model.

**Linear sequence extension:** A linear sequence extension is a topological sort that defines an order in which to learn these course objects such that each is performed only after all of its prerequisite constraints are satisfied. A linear sequence extension orders the vertices and edges of a directed acyclic graph in a simple and consistent way and hence plays the same role as the path graph in linear time. This topology is very valuable for programmers in developing course authoring programs.

Three important facts about linear sequence extensions are:

- Only path graphs can have linear sequence extensions, since any directed cycle is an inherent contradiction to a linear ordering of the course objects.
- Every path graph can be a linear sequence extension, so there must always be at least one schedule for any set of reasonable precedence constraints among course objects.
- A linear sequence extension allows many such schedules, especially when there are few constraints. Consider  $n$  nodes without any constraints. Any of the  $n!$  permutations of the course objects constitutes a valid linear sequence extension.

**Node and edges:** An object in the concept path graph or linear sequence extension is called a “node” and a line drawn between nodes is called an “edge.” Edges represent relationships between nodes.

#### 4.2. Generation of the Concept Path Graphs

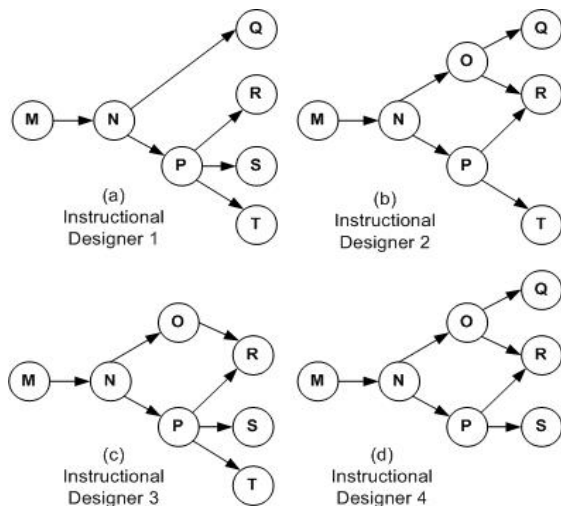


Figure 2. The concept graphs of four designers

In Section 4.1 we gave a definition of the path graphs. In the path graph combination model, we arrange course objects in an order consistent with the domain model and generate the concept path graphs using the learning sequence just defined. Figures 2(a), 2(b), 2(c) and 2(d) present the content path graphs defined by four instructional designers.

#### 4.3. The Concept Graph Combination Model

Based upon the example in Figure 2, we present an algorithm to combine the different path graphs into a single pattern. The algorithm steps are illustrated by the example shown in Figures 2(b) and 2(c) and described in detail below.

##### Step 1: Linear Sequence Extension Generating.

A linear sequence extension of a content path graph is easily found in linear time. We define the set of nodes as the same as the nodes that appeared in content path graph. Topological sorting orders the nodes and the directed edges inherited from the content path by using the linear sequencing.

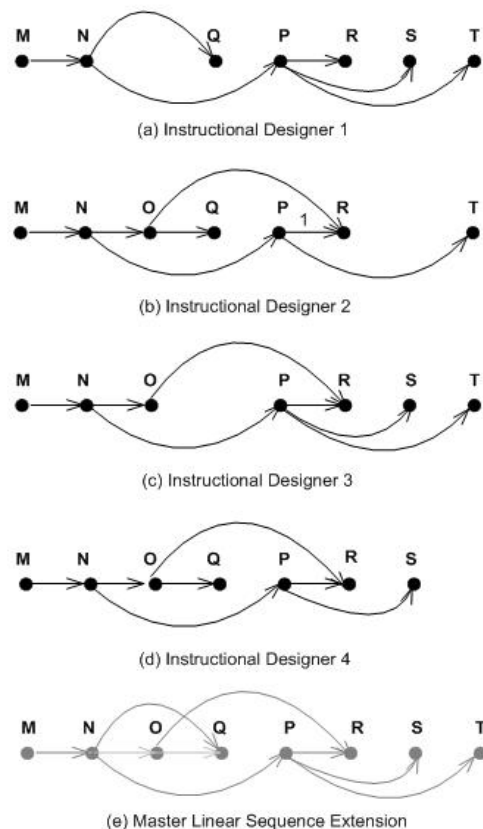


Figure 3. The linear sequence extensions

### Step 2: Closeness index calculation

Goldsmith's method is used to calculate the closeness index and ordering processes [1]. We repeat this method for every pair of instructional designers. The method is explained as follows.

- Determine the set of nodes L  
e.g. L = <M, N, O, P, Q, R, S, T>
- For each item belonging to L
  - Consider the linear sequence extension from instructional designer number j (ID<sub>j</sub>) and instructional designer number k (ID<sub>k</sub>).
  - For every target node of each instructional designer, insert the related node into the related set.
  - Calculate the Intersection Set (IS<sub>jk</sub>), and Union Set (US<sub>jk</sub>).
  - Calculate the closeness coefficient (CC<sub>jk</sub>) with equation (1).

$$CC_{jk} = \frac{|IS_{jk}|}{|US_{jk}|} \quad (1)$$

where

|US<sub>jk</sub>| is the number of members in the Union Set

|IS<sub>jk</sub>| is the number of members in the Intersection Set

For example, the comparison between ID<sub>2</sub> and ID<sub>3</sub> is derived from the example in Figure 3(c) and 3(d). The closeness coefficient (CC<sub>23</sub>) is shown in Table 1. We repeat this process for every pair of instructional designers and the overall results are shown in Table 2.

**Table 1. Comparison results between two instructional designers (designer 2 and designer 3)**

Node	ID 2	ID 3	US <sub>23</sub>	IS <sub>23</sub>	CC <sub>23</sub>
M	{N}	{N}	{N}	{N}	1.00
N	{M,O,P}	{M,O,P}	{M,O,P}	{M,O,P}	1.00
O	{N,Q,R}	{N,R}	{N,Q,R}	{N,R}	0.67
P	{N,R,T}	{N,Q,S,T}	{N,Q,R,S,T}	{N,T}	0.40
Q	{O}	{O,P}	{O,P}	{P}	0.50
R	{O,P}	{P}	{O,P}	{P}	0.50
S	∅	{P}	{P}	∅	0.00
T	{P}	{P}	{P}	{P}	1.00
Σ CC <sub>23</sub>					<b>5.07</b>

**Table 2. Results of closeness coefficient calculation**

Node	ID1-ID2	ID1-ID3	ID1-ID4	ID2-ID3	ID2-ID4	ID3-ID4
M	1.00	1.00	1.00	1.00	1.00	1.00
N	0.50	0.50	0.50	1.00	1.00	1.00
O	1.00	0.00	0.00	0.67	1.00	0.67
P	0.75	0.60	0.75	0.40	0.50	0.40
Q	0.50	0.00	0.00	0.50	1.00	0.50
R	0.50	1.00	0.50	0.50	1.00	0.50
S	0.00	1.00	1.00	0.00	0.00	1.00
T	0.00	1.00	0.00	1.00	0.00	0.00
Σ CC <sub>jk</sub>	<b>4.25</b>	<b>5.10</b>	<b>3.75</b>	<b>5.07</b>	<b>5.50</b>	<b>5.07</b>

- Calculate the closeness index between ID<sub>j</sub> and ID<sub>k</sub>, C(ID<sub>j</sub>,ID<sub>k</sub>) with equation(2),  
e.g. C(ID<sub>2</sub>,ID<sub>3</sub>) = (1/8)\*(5.07) = 0.6338.

$$C(ID_j, ID_k) = \frac{1}{|L|} \sum CC_{jk} \quad (2)$$

where |L| is the number of items in L.

### Step 3: Confidence value calculation

We compute the confidence values of instructional designers by using the closeness index from step 2 with equation (4). The results are shown in Table 3.

$$\text{Total}(ID_j) = \sum_{k=1}^m C(ID_j, ID_k) \quad (3)$$

$$\text{Confidence}(ID_j) = \frac{\text{Total}(ID_j)}{\sum_{i=1}^m \text{Total}(ID_i)} \quad (4)$$

where m is the number of instructional designers

**Table 3. Results of the confidence value calculations**

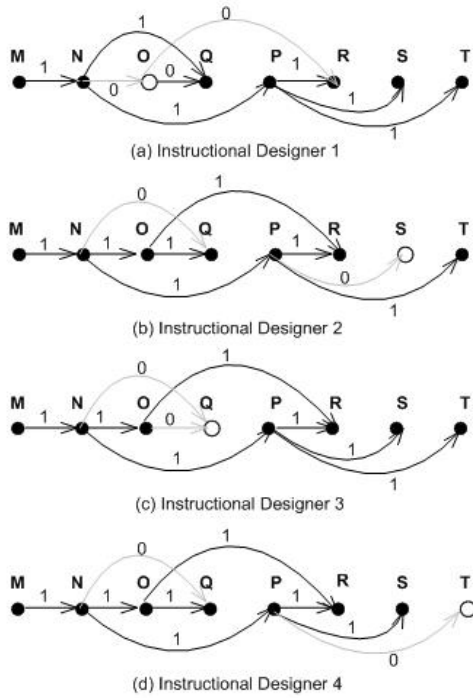
ID#	ID 1	ID 2	ID 3	ID 4
ID 1		0.5313	0.6375	0.4688
ID 2	0.5313		0.6338	0.6875
ID 3	0.6375	0.6338		0.6333
ID 4	0.4688	0.6875	0.6333	
Total	<b>1.6375</b>	<b>1.8525</b>	<b>1.9046</b>	<b>1.7896</b>
Confidence	<b>0.2279</b>	<b>0.2579</b>	<b>0.2651</b>	<b>0.2491</b>

#### Step 4: Integrating the Linear Sequence Extension

The steps in the linear sequence extension combination mechanism are explained below.

**Step 4.1:** Define the node weight and edge weight value of each relation between nodes. We consider two entities: Nodes and edges.

In the weighting process of the linear sequence extension, the black nodes represent the existing nodes. On the other hand, the white nodes represent the nonexistent nodes. We represent the existing edges as black lines and the nonexistent edges as gray lines. The results of weighting process are shown in Figure 4.



**Figure 4. The Results of Weighting Process**

In the case of nodes, we define the weight value as 1 for existing nodes and 0 for nonexistent nodes. Similarly in case of an edge, we define the weight value as 1 for existing edges and 0 for nonexistent edges. Equation (5) is used to calculate the total weight of each node and edge. We call the existing nodes or existing edges existing values in this equation. The important factor in the total weight is the instructional designer's confidence. If the total weight is larger than 0.6, nodes or edges under consideration are kept. The results of both cases appear in Tables 4 and 5.

Total Node(or Link) Weight =

$$\Sigma \left( \text{existing\_value} * \text{Confidence} \left( D_j \right) \right) \quad (6)$$

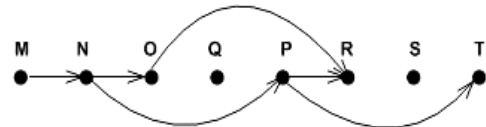
**Step 4.2:** Using the results mentioned above, two edges; N-Q and O-Q, are deleted from master linear sequence extension (see Figure 3). The integrated linear sequence extension is shown in Figure 5.

**Table 4. The weighted values of nodes**

	ID 1 (0.2279)	ID 2 (0.2579)	ID 3 (0.2651)	ID 4 (0.2491)	Total Node Weight				
M	1	0.2279	1	0.2579	1	0.2651	1	0.2491	<b>1.0000</b>
N	1	0.2279	1	0.2579	1	0.2651	1	0.2491	<b>1.0000</b>
O	0	0.0000	1	0.2579	1	0.2651	1	0.2491	<b>0.7721</b>
P	1	0.2279	1	0.2579	1	0.2651	1	0.2491	<b>1.0000</b>
Q	1	0.2279	1	0.2579	0	0.0000	1	0.2491	<b>0.7349</b>
R	1	0.2279	1	0.2579	1	0.2651	1	0.2491	<b>1.0000</b>
S	1	0.2279	0	0.0000	1	0.2651	1	0.2491	<b>0.7421</b>
T	1	0.2279	1	0.2579	1	0.2651	0	0.0000	<b>0.7509</b>

**Table 5. The weighted values of links**

	ID 1 (0.2279)	ID 2 (0.2579)	ID 3 (0.2651)	ID 4 (0.2491)	Total Link Weight				
M-N	1	0.2279	1	0.2579	1	0.2651	1	0.2491	<b>1.0000</b>
N-O	0	0.0000	1	0.2579	1	0.2651	1	0.2491	<b>0.7721</b>
N-Q	1	0.2279	0	0.0000	0	0.0000	0	0.0000	<b>0.2279</b>
N-P	1	0.2279	1	0.2579	1	0.2651	1	0.2491	<b>1.0000</b>
O-Q	0	0.0000	1	0.2579	0	0.0000	1	0.2491	<b>0.5070</b>
Q-R	0	0.0000	1	0.2579	1	0.2651	1	0.2491	<b>0.7721</b>
P-R	1	0.2279	1	0.2579	1	0.2651	1	0.2491	<b>1.0000</b>
P-S	1	0.2279	0	0.0000	1	0.2651	1	0.2491	<b>0.7421</b>
P-T	1	0.2279	1	0.2579	1	0.2651	0	0.0000	<b>0.7509</b>

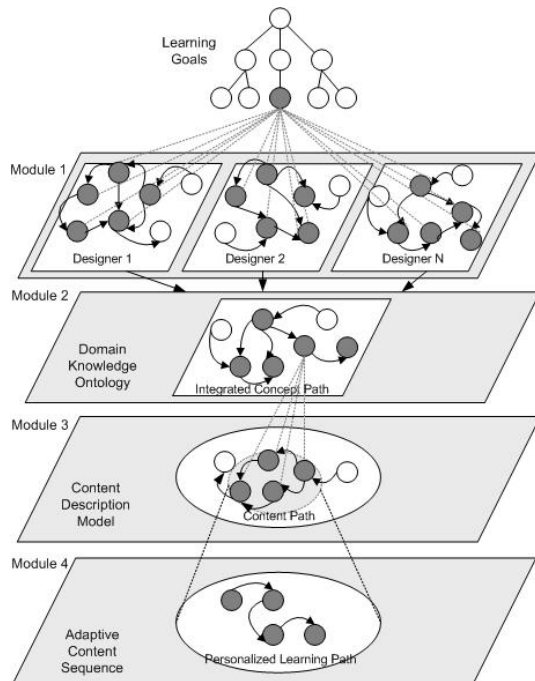


**Figure 5. The integrated linear sequence extension**

## 5. Integrated Content Path Combination Model for Generating Personalized Learning Paths

Our personalized learning path model contains four main modules that carry out the linear extension combination algorithm.

The linear extension combination algorithm is used in the concept path combination model (CPCM) that is one of the modules of our personalized learning path model shown in Figure 6. The algorithm aims to provide a solution for various design problems and to produce a strongly recommended concept sequence path that is based on the domain model and the user model.



**Figure 6. The content path combination model**

Figure 6 represents the overall model for generating the personalized learning paths. We explain the detail of each module as follows:

- **Module 1: Generating the concept path**, providing the different concept paths that are designed by each instructional designer. The concept path is designed based on the learner knowledge space and must be consistent with the learning goals.

- **Module 2: Generating the integrated concept path**, finding the closeness index, confidence value and relationships among the designs produced by different designers. These factors are used to find the integrated concept path following by the concept path combination model. In this paper, we focused on this module.
- **Module 3: Generating the content path**, using the integrated concept path to find the suitable content (course object) based on the content description model. The course object must be consistent with the concept path.
- **Module 4: Generating the personalized learning path**, providing the mechanism for finding the most suitable learning path for each learner based on the adaptive content sequence model.

All of these modules use learning goals that specify the objectives and conditions for satisfying the course requirements. The result obtained from the concept path combination model is an integrated concept path. It is to be used to recommend a sequence of concepts for making a pattern to the content path generation module. In addition, the concept path combination model scopes the area of wide designs and can be used in the same way in every level of course unit, whether it is the lesson level, the topic level, or the subtopic level.

## 6. Conclusions and Future Work

We have proposed a new concept path combination model that handles the different designs of various course designers and that is capable of supporting the personalize learning paths in the future. This model includes an algorithm called “CPCM” that is based on linear sequence extensions, our closeness index and confidence value calculations. The styles of four different instructional designers are used to illustrate the processes of finding the relationships among designers, calculating the designer’s confidence value, combining various concept path designs, and building an integrated linear sequence extension.

We plan to build on our previous work [8] using the integrated linear sequence extension obtained from CPCM to determine the most suitable order of the course content based on the content description model.

We also plan to develop an estimation function to be used for finding the most suitable course objects for satisfying the concept path graph and the learning goals. To support adaptive E-learning systems, we aim to implement personalized learning paths, based on our conviction that any path among courses can be better chosen by examining the characteristics of different learners and the requirements of instructional designers.

## References

- [1] H. Y. Shyu, S. H. Hsieh, and Y. H. Chou, "Integrating Concept Maps into Designing a Courseware Management System," In *Proceedings of the Learning Objects 2003 Symposium*, Honolulu, Hawaii, USA, June 2003, pp. 51-60.
- [2] IEEE (2002), "*IEEE Standard for Learning Object Metadata*," 1484.12.1-2002, 2002.
- [3] K. Frankolu, "Why Online learners drop out," *Workforce*, June 2001, pp. 53-63.
- [4] M. Andric, V. Devedzic, W. Hall and L. Carr, "à la in Education: Keywords Linking Method for Selecting Web Resources", In: C. K. Looi, G. McCalla, B. Bredeweg and J. Breuker (eds.), *Artificial Intelligence in Education*, IOS Press, Amsterdam, Netherland, 2005, pp. 25-32.
- [5] M. Kravcik and M. Specht, "Flexible Navigation Support in the WINDS Learning Environment for Architecture and Design," In *Proceedings of the Third International Adaptive Hypermedia and Adaptive Web- based Systems Conference*, August 23-26, 2004, Eindhoven, The Netherland, pp.156-165.
- [6] N. Henze and W. Nejd, "Adaptation in Open Corpus Hypermedia," *International Journal of Artificial Intelligence in Education*, Vol.12, Iss. 4, 2001, pp. 325-350.
- [7] N. Henze, "Personal Readers: Personalized Learning Object Readers for the Semantic Web," In: C. K. Looi, G. McCalla, B. Bredeweg and J. Breuker (eds.), *Artificial Intelligence in Education*, IOS Press, Amsterdam, Netherland, pp. 274-281.
- [8] N. Pukkhem and W. Vatanawood, "A Multi-Instructor Cooperative Model for Supporting Learning Objects Aggregation based on XML-Based Planning Strategy," *WSEAS Transactions on Computer*, Vol. 4, Iss. 10, October 2005, pp. 1390-1398.
- [9] N. Pukkhem and W. Vatanawood, "Instructional Design using Component-Based Development and Learning Object Classification," In *Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies*, Taiwan, July 2005, pp. 492-494.
- [10] P. Brusilovsky, J. Eklund and E. Schwavz, "Web-based Education for All: A Tool for Developing Adaptive courseware," *Computer Networks and ISDN Systems*, Vol. 30, 1-7, 1998, pp. 291-300.
- [11] P. Brusilovsky S. Sosnovsky, M. Yudelson and G. Chavan, In: C. K. Looi, G. McCalla, B. Bredeweg and J. Breuker (eds.), "Interactive Authoring Support for Adaptive Educational Systems," *Artificial Intelligence in Education*, IOS Press, Amsterdam, Netherland, 2005, pp. 96-103.
- [12] P. De Bra, G. J. Houben and H. Wu, "AHAM: A Dexter-Based Reference Model for Adaptive Hypermedia," In *Proceedings of the ACM Conference on Hypertext and Hypermedia*, February 21-25, 1999, Darmstadt, Germany, pp. 147-156.
- [13] P. De Bra, L. Aroyo and A. Cristea, "Adaptive Web-based Educational Hypermedia." *Web Dynamics, Adaptive to Change in Content, Size, Topology and Use*, 2004, Heidelberg, Germany: Springer, pp. 387-410.
- [14] P. Doloy, N. Henze, W. Nejd and M. Sintex, "The Personal Reader: Personalizing and Enriching Learning Resource Using Semantic Web Technologies," In *Proceedings of the Third International Adaptive Hypermedia and Adaptive Web-based Systems Conference*, August 23-26, 2004, Eindhoven, The Netherland, pp. 85-94.
- [15] W. Dick, L. Carey, and J. O. Carey, *The Systematic Design of Instruction*, 6<sup>th</sup> edition, 2005, MA, Boston, USA: Allyn & Bacon, pp. 183-196.