An Adaptive Course Generation Framework

Frederick W.B. Li[1], Rynson W.H. Lau[2], and Parthiban Dharmendran[1]

[1] School of Engineering and Computing Sciences, Durham University, United Kingdom
[2] Department of Computer Science, City University of Hong Kong, Hong Kong

ABSTRACT: *Existing adaptive e-learning methods are supported by student (user) profiling for capturing student characteristics, and course structuring for organizing learning materials according to topics and levels of difficulties. Adaptive courses are then generated by extracting materials from the course structure to match the criteria specified in the student profiles. In addition, to handle advanced student characteristics, such as learning styles, course material annotation and programming-based decision rules are typically used. However, these additives demand certain programming skills from an instructor to proceed with course construction; they may also require building multiple course structures to handle practical pedagogical needs, which may be complicated to develop and maintain.*

In this paper, we propose a framework based on the concept space and the concept filters to support adaptive course generation where comprehensive student characteristics are considered. The concept space is a data structure for modeling student and course characteristics, while the concept filters are modifiers to determine how the course should be delivered. They offer a simple and unified way for modeling and processing a variety of student learning needs as well as different factors that affect course material relevance. Because of the "building block" nature of the concept nodes and the concept filters, the proposed framework is extensible. More importantly, our framework does not require instructors to equip with any programming skills when they construct adaptive e-learning course.

Keywords: User profiling, student profiles, course profiles, resource profiles, adaptive e-Learning.

INTRODUCTION

e-Learning is a technology supported learning approach, where the students' learning activities are assisted with communication and multimedia technologies (Li et al., 2008). This provides students with virtually unlimited access to knowledge and improves their learning through multi-modality materials. In addition, learning may also be adapted to individual paces, allowing students to learn at any time and place to match their own needs. On top of this, student (user) profiling can be added to capture student characteristics, such as learning preferences, background knowledge and learning progress, to help generate tailored learning materials and support adaptive e-learning.

Early work, such as InterBook (Brusilovsky et al., 1998), utilizes a hierarchical structure to organize course materials according to the topics and levels of difficulties, and uses student profiles as matching criteria to extract tailored learning materials from the course structure to produce adaptive e-learning courses. Recent work (Wu et al., 2001, De Bra et al., 2003, Stash et al., 2004) resorts the adaptive course generation to course material annotation and programming-based decision rules. They facilitate the generation of adaptive courses for students with different learning styles, such as example-oriented or activity-oriented learners (Honey et al., 1992) by selecting appropriate type of course materials and presenting them in a desired sequence. However, such methods demand more technical skills from an instructor for constructing adaptive courses. In addition, it is not straight-forward to apply such methods to handle certain practical pedagogical needs, such as constructing a course for students with very different academic backgrounds, balancing learning workload across course aspects when accommodating student learning preferences, and dynamically adjusting the depth of a course topic for delivery based on its popularity or other factors.

To allow ordinary instructors constructing adaptive courses, which account a variety of pedagogical needs, without acquiring prior programming knowledge or relying on technical assistance, we have developed a framework based on the concept space and the concept filters. The concept space is a data structure for modeling student and course characteristics, while the concept filters are modifiers to determine how the course should be disseminated. Based on them, we also provide a unified three-tier profiling mechanism, which comprises student, course and resource profiles, to facilitate the adaptive course generation. The rest of this paper is organized as follows. Section 2 gives a survey of related work. Section 3 presents the proposed framework. Section 4 evaluates the proposed framework through a number of experiments. Finally, Section 5 briefly concludes the work presented in this paper.

RELATED WORK

A learning process is driven by “what to learn”, i.e., the scope of learning, and “how to learn”, i.e., how a student approaches such learning scope. Adaptive e-learning addresses these two questions by offering students with tailored learning materials. Existing work on adaptive e-learning tackles this problem by applying student profiles on well organized courseware. More specifically, a student profile captures the learning preferences, background knowledge/experiences and learning progress of the student. It forms the basis for filtering a pool of course materials to pick out relevant ones. For instance, InterBook (Brusilovsky et al., 1998) organizes course materials in a hierarchical structure along with indices according to the topics and level of difficulties. (Middleton et al., 1998) improves the discovery of relevant course materials by knowledge classification based on ontology and collaborative choices made by a group of users. The ontology (Studer et al., 1998) formulates the grouping and the relation among concepts. It is commonly applied to organize course materials and to form the metric for determining the user required materials. Another example can be found in (Dolog et al., 2004). The utilization of collaborative information (Balabanović et al., 1997) can enhance the accuracy of the retrieved course materials, as it complements the incompleteness or impreciseness of individual user profiles. (Freyne et al., 2007) has exploited user browsing and searching patterns to give more precise modeling on collaborative information. All of the above methods focus on addressing the “what to learn” problem.

The “how to learn” problem is also crucial to adaptive e-learning. Student learning styles (Felder et al., 1988) in terms of psychological features, such as sequential, global, active and reflective, are considered as a key to address this problem. Ways to acquire and understand student learning styles have been proposed by (Schiaffino et al., 2008). The implication of learning style is that a student may need to perform different tasks or follow a different sequence and abstraction level of learning materials in order to understand a piece of concept. (Papanikolaou et al., 2003) applies learning styles to allow students to learn a concept through different interaction styles, such as theory-oriented, exercise-oriented or activity-oriented. MOT+AHA! (Stash et al., 2004) supports more types of learning styles by course material annotation and programming-based decision rules. This is undeniably more powerful. However, it demands more technical skills from an instructor or relies on technical assistance for constructing adaptive courses, which is practically unfavorable. In addition, it is not straight-forward to apply such a method to handle complicated but practical pedagogical needs, such as constructing a course for students with very different academic backgrounds, balancing learning workload across course aspects when student learning preferences are accounted, and adjusting the depth of a course topic for delivery based on the dynamic nature of course material relevance and the reference material exploration. Note that the relevance of course materials depends not only on the needs of individual students or an entire cohort, but also on some dynamic factors such as popularity, maturity, stability, features and innovativeness of the course materials (Yue et al., 2004).

On the other hand, students usually learn independently in an e-learning environment and use Web searching as a popular means for problem solving and information exploration. The main concern here is the students' ability to identify relevant reference materials. Undeniably, PageRank (Page et al., 1998), which examines the entire link structure of the Web to determine the importance of each Web page, is the most popular technique to this searching problem. (Qiu et al., 2006) extends PageRank by generating topic specific rankings of Web pages and allows matching such rankings against user profiles to obtain tailored search results. This partially addresses the reference material exploration problem in adaptive e-learning. However, we note that both the student learning style and the relevance of course materials due to external factors (Yue et al., 2004) play crucial roles in determining the suitability of reference materials, but they have not been properly addressed.

OUR FRAMEWORK

Our research on adaptive e-learning leads to a very interesting finding. Existing work supports adaptive e-learning mainly through student-profiling, which requires a dedicated course material structure. However, we note that the definitions of many important parameters that support adaptive e-learning, such as different aspects of learning styles, cohort (or group) learning preferences and course content related factors, are not set in the student profiles; they should be defined together with individual courses. Hence, we have developed a framework based on the *concept space* and the *concept filters*. The concept space is a data structure for modeling student and course characteristics, while the concept filters are modifiers to determine how the course should be disseminated. Based on them, we also provide a unified three-tier profiling mechanism, which comprises student, course and resource profiles, to facilitate the adaptive course generation.

Concept Space

A *concept space* K is defined as a confined set of knowledge. As shown in Figure 1, it is modelled as a collection of *concept nodes* k_n. Each k_n represents a small piece of knowledge and is uniquely identified with an index *n*. It is modelled as a multi-dimensional space, where each dimension is referred to as an *aspect* of the knowledge that k_n represents. The aspects can be defined based on ontology or the expertise of a course designer. For example, a concept node “content management and layout design” may constitute a number of aspects: “content management system (CMS)”, “layout design applications” and “CSS programming”, etc.. Each aspect of a concept node attaches two indicators, an *aspect weight* and a *level of abstraction*, which are defined with a scale between 0 and 1. The aspect weight indicates the importance / relevance of an aspect, while the level of abstraction describes how detailed the corresponding learning materials of an aspect should be released. In addition, *aspect groups* can be set up to group related aspects. We have defined three aspect groups in our current implementation, namely the *interaction style group*, *content style group* and the *polysemy group*, which handle the learning activities, learning approaches and the multiple meanings of a concept, respectively. We allow a course designer to define a course using concept spaces rather than through constructing a complicated ontology hierarchy. This significantly simplifies the process of creating adaptive e-learning courses, making it possible for general instructors to create these courses without technical assistance.

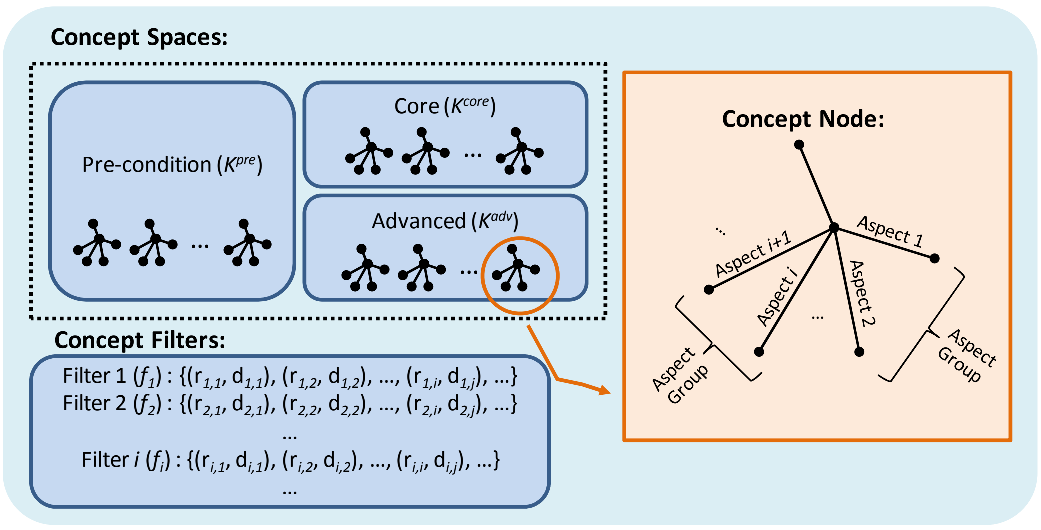

Concept Spaces:
Pre-condition (K^{pre})
Core (K^{core})
Advanced (K^{adv})
...
Concept Filters:
Filter 1 (f_1) : {($r_{1,1}$, $d_{1,1}$), ($r_{1,2}$, $d_{1,2}$), ..., ($r_{1,i}$, $d_{1,j}$), ...}
Filter 2 (f_2) : {($r_{2,1}$, $d_{2,1}$), ($r_{2,2}$, $d_{2,2}$), ..., ($r_{2,i}$, $d_{2,j}$), ...}
...
Filter *i* (f_i) : {($r_{i,1}$, $d_{i,1}$), ($r_{i,2}$, $d_{i,2}$), ..., ($r_{i,i}$, $d_{i,j}$), ...}
...
Concept Node:
Aspect 1
Aspect 2
Aspect i
Aspect i+1
Aspect Group
Aspect Group

Figure 1: The unified profile structure.

Concept Filters

Concept node filtering is a mechanism to shape a concept space to fulfil learning needs from different students. A concept filter comprises filters at different concept node levels, i.e., *concept*, *aspect group* and *aspect* levels. All types of filters are formulated in the same way as a list of (r_i, d_i)-pairs, where r_i and d_i are the weight for adjusting the importance and the abstraction level, respectively, of an element at a concept node level and both of them are defined with a scale between 0 and 1. A (0, 0)-pair means that the corresponding element should be ignored. While the weight indicates the importance of each element at a concept node level, the abstraction level indicates the degree of detail of the course material associated with the element to be presented to the student. In our implementation, we use concept filters to formulate individual / collaborative learning preferences, the learning / content / interaction styles and the external factors. However, the concept filters are not limited to these usages. Any learning needs that require presenting learning materials with a preferred order or abstraction level may also be modelled using concept filters.

The idea of the concept filter is an important feature of our framework. It provides course designers a very handy tool for developing adaptive e-learning courses that take into account comprehensive student characteristics. Here, we discuss the construction of the concept filters. We do not attempt to illustrate the filter construction for all possible student characteristics. Instead, by discussing the filters used in our experiments, we lay out their design considerations, which set a reference for course designers to design other filters.

Content Styles: They describe how the learning materials may be stylized to fit a student's background so that the student may understand the course better. For example, a student without a computer science background may prefer to learn Web site construction using a non-programming approach. This requirement may not be well addressed by simply using ontology to extract the non-programming based learning materials, as this student may still need to learn some relevant programming concepts in order to meet the learning outcome requirements. Hence, we need to disseminate not only the non-programming based materials but also some essential programming based materials. With our framework, we setup content style filters using *aspect group level filters*, where each (r_i, d_i)-pair in a filter is set to adjust the importance and the abstraction level of an aspect group. In addition, instead of using separate filters to process each concept node, we may setup a combined filter to process the aspect groups of all concept nodes in a course as a whole. For instance, suppose that we have a course with 3 concept nodes, where each concept node comprises 2 aspect groups. To handle students that may learn better if the course focuses more on the first aspect group of each concept node, the content style filter may be set in this form: {(1, L_1), (1, S_1), (1, L_2), (0, 0), (1, L_3), (1, S_3)}, where L_i represents a large value and S_i represents a small one. For example, the first two (r_i, d_i)-pairs show that the first and the second aspect groups of concept node 1 need to be disseminated in a high and low details, respectively. The (0, 0)-pair in the filter shows the second aspect group of concept node 2 needs not to be disseminated.

Learning Styles: They describe how students may change the order and level of detail of the subject topics in learning a course. For instance, students with the *global learning style* can understand a course easier by studying some high level concepts to get an overview of the course first before going into the details. Students with the *sequential learning style* need to finish a topic before they can begin another one. To set the *global learning style filters*, we may initially assign small values to all level-of-abstraction items to extract a high level abstraction of course materials from each concept node. Such filters can be setup using *aspect level filters* to allow a precise control on the dissemination of individual aspects of each concept node. After a student has finished this learning stage, it will be marked in the student profile as a part of the prior knowledge. The learning goal will also be revised by taking out this prior knowledge. We then apply another global filter with larger values in the level-of-abstraction items to deliver students with more course details. This filtering and student profile updating processes are repeated until all relevant course contents have been delivered.

Three-Tier Profiling

As shown Figure 2, the three-tier profiling mechanism comprises *student*, *course* and *resource* profiles. A student profile stores the student background and the on-going learning scope. It also maintains records on how the student approaches a course. This student profile is course specific. The on-going learning scope is updated at each learning stage of a student to reflect his/her learning progress. A course profile helps maintain the course structure according to major subject topics and provides information on how students may learn the course with respect to a variety of learning, content and interaction styles. A resource profile models learning resources (Lehmann et al., 2008) authored by both the instructor and the peer students, where the instructor offers the course specific assumptions or pre-conditions while the students develop course related preferences based on their learning experiences. This resource profile offers informal help to support student learning of a particular course or across the whole discipline.

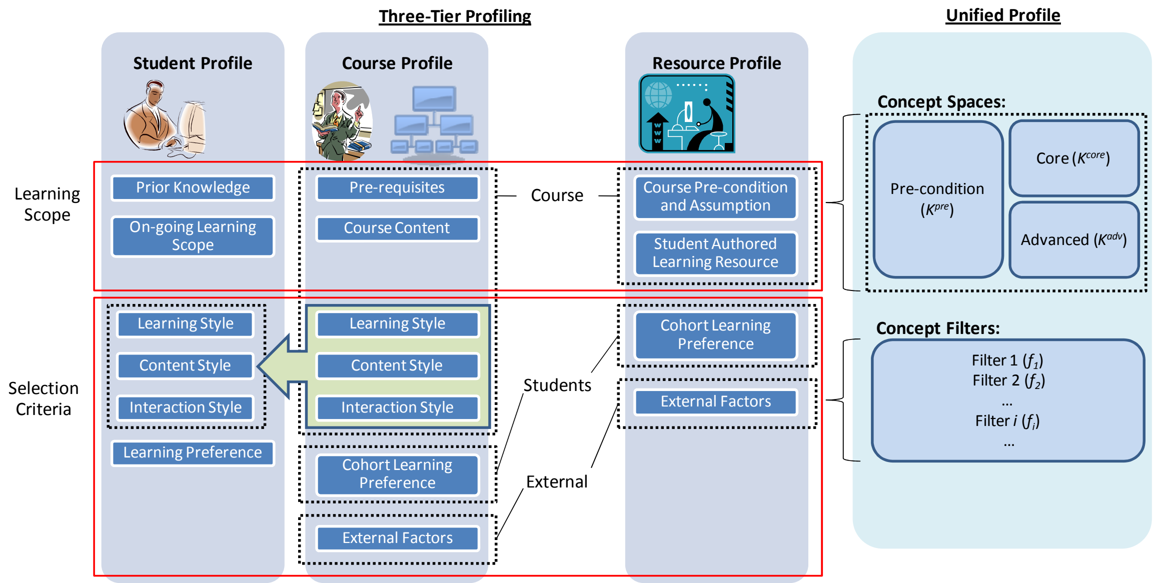


Figure 2: Conceptual model of the three-tier profiling mechanism.

Unified Profile Structure

Although the information maintained in the three types of profiles are significantly different, we note that the information within each profile can be categorized into learning scope and selection criteria for defining “what to learn” and the ways to determine it. Hence, we use a concept space structure and a concept filter set to formulate these two types of information, and let them form a *unified profile* structure, as shown in Figure 1. The unified profile structure is composed of three *concept spaces*, the pre-condition K^{pre}, core K^{core} and advanced K^{adv} concept spaces, plus a *concept filter set* F. Each of the concept spaces is modeled as a collection of *concept nodes*, which will be discussed in next sub-section. K^{pre} defines the prior requirements before tailor-made learning content can be determined, while K^{core} and K^{adv} define the core and the advanced learning contents. The concept filter set F is used to modify a concept space through adjusting the importance and the level of abstraction of each concept node inside the concept space.

To form a *student profile* with this unified profile structure, K^{pre} defines the prior knowledge or experiences of a student, while K^{core} and K^{adv} give the core and advanced learning scopes of the student in a course context. These learning scopes are updated at each learning stage to reflect the student's learning progress. The actual concept filter set *F* of these styles, which is used to generate adaptive courses, is stored in the course profile. It includes filters to handle the course specific learning preferences and a variety of learning, content and interaction styles. Note that the learning characteristics of a student can be captured through analyzing how the student deals with different learning activities (Schiaffino et al., 2008).

To form a *course profile* with the unified profile structure, K^{pre} defines the pre-requisite or the entrance requirement of the course, while K^{core} and K^{adv} define the core and advanced learning scopes of the course. The concept filter set F includes filters to handle learning styles (Felder et al., 1988), content styles and interaction styles (Papanikolaou et al., 2003), which define how the course content can be matched with these styles. In addition, to allow an instructor to take into account meaningful student opinions for course design enhancement, the course profile also includes filters to describe the collaborative learning preferences of previous students. Finally, it also comprises filters to define the importance or relevance of each piece of course content against the external factors (Yue et al., 2004), such as popularity and maturity of the course materials.

In contrast to the traditional learning environment, students in an e-learning environment may engage in learning more independently. The *resource profile* is designed to support such a need. It stores two types of information. First, it includes course specific assumptions or pre-conditions set by an instructor, which comprise definitions of the course or discipline specific terminologies and the relationship among them. As an example, we may define the course related synonym and the polysemy in a resource profile. Second, the resource profile also includes student authored learning resources (Lehmann et al., 2008) from peer students. These two types of information are formulated into concept nodes in the way that K^{pre} defines the set of concepts for exclusion, while K^{core} and K^{adv} give the core and the advanced concept definitions and relationships. The concept filter set F here includes filters to take care of the cohort learning preferences and the external factors that affect the concept relevance of the course.

The difference between a course profile and a resource profile is that a course profile is set by the instructor for modelling the relation among major subject topics, while a resource profile allows both the instructor and peer students to collaboratively develop learning resources in terms of the course or discipline specific terminologies to aid independent learning. Pedagogically, a course profile formally defines how the instructor teaches a course, while a resource profile offers informal help to support student learning of a particular course or across the whole discipline. With the resource profile in place, when a student performs a Web search for reference materials, the query may be refined by the resource profile to help obtain better fit results, which match the context of the course or the whole discipline.

EXPERIMENTAL RESULTS AND EVALUATIONS

We have conducted a number of experiments to demonstrate the generation of adaptive course notes to different students based on their characteristics. The experiments were conducted on the “Web Technologies” course offered to 25 students. The course design as shown in Table 1 is set by the course instructor. It lays out the definitions of the core and advanced concept nodes, the aspect groups and the aspects of each concept node, and the weight and the level of abstraction of each aspect. Note that the aspect groups are set up based on the *programming* and the *non-programming* types of content styles to indicate the approach that a course aspect will be disseminated to students.

Aspect Group	Course Content		Aspect weight	Level of abstraction
	Content management and layout design (concept node 1)			
Non-programming **(AG1.1)**	Content management systems [CMS] **(A1.1)**	Core	0.8	0.8
		Advanced	0.8	0.4
	Layout design application **(A1.2)**	Core	0.4	0.6
		Advanced	0.4	0.4
Programming **(AG1.2)**	CSS Programming **(A1.3)**	Core	0.8	0.6
		Advanced	0.8	0.4
	Web site construction (concept node 2)			
Non-programming **(AG2.1)**	Webpage Basics [HTML] **(A2.1)**	Core	0.8	0.8
		Advanced	0.6	0.4
	Web site building tools [Dream weaver] **(A2.2)**	Core	0.8	0.8
		Advanced	0.8	0.6
Programming **(AG2.2)**	Web Applications - Ruby on Rails **(A 2.3)**	Core	0.6	0.6
		Advanced	0.6	0.2
	Data management for dynamic Web applications (concept node 3)			
Non-programming **(AG3.1)**	MS Access Database **(A3.1)**	Core	0.4	0.6
		Advanced	0.6	0.4
Programming **(AG3.2)**	MySQL Database **(A3.2)**	Core	0.6	0.8
		Advanced	0.4	0.2

Table 1: The course design of the Web technology course.

The students taken the course had a bachelor degree in either computer science (CS students) or other discipline (non-CS students). Their profiles are shown in Tables 2 and 3. The profiles store the students’ course relevant *prior knowledge*, *learning styles* that indicate their intrinsic and psychological features, and *learning preferences* that indicate their favorite topics of the course. The prior knowledge of each student on a course aspect is described with a scale between 0 (no knowledge) and 1 (complete knowledge). In addition, a student scored 0.6 or above on the prior knowledge of a course aspect means that the student has already gained an “above average” knowledge level on this course aspect; otherwise, he/she is considered as not having sufficient knowledge. To apply the students’ prior knowledge, a student was exempted from studying the core part of course aspect(s) if he/she had already obtained an above average knowledge level. For example, students S19 and S25 were exempted from studying the core part of Webpage Basics [HTML] (A2.1), but still needed to take the advanced part of it. To show what adaptive courses are produced for different students, we present and discuss the results here in groups based on the similarity of student characteristics.

Non-CS Students with Experience in MS Access Database (Group A)

As shown in Table 2, students S1, S2, S8, S13, S15 and S18 are non-CS students with above average knowledge in MS Access Database (A3.1). To let these students feel comfortable in learning such a technology-based course while freeing them from spending time to study the introductory materials of MS Access Database, the course was tailor-made to these students by considering their *prior-knowledge* and intrinsic characteristics.

Prior-Knowledge and Content Style: The changes in course dissemination are shown in Figure 3. There are three bars associating with each course aspect to show the level of abstraction (LOA) to be delivered for that course aspect when a different student characteristic is applied. The first bar of each aspect indicates the deliverable LOA of that aspect as prescribed in the course design (Ref. Table 2). The second and the third bars show the changes in the aspect's LOA after the students' prior-knowledge and content style have been applied, respectively. Note that each bar comprises two parts. The darker part represents the *core* part of the course aspect, while the lighter one represents the *advanced* part of the course aspect. This convention is also applied to Figures 5, 7 and 9.

Concept Space	**Students**																	
Prior Knowledge	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**	**S18**
Content management systems [CMS] (**A1.1**)	0	0	0	0	0	0	0.4	0	0	0.4	0.6	0	0	0	0	0	0	0
Layout design application (**A1.2**)	0.2	0.2	0	0	0	0	0.6	0	0	0.6	0.6	0	0	0	0.4	0	0	0.2
CSS Programming (**A1.3**)	0.4	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0	0	0.2
HTML Programming (**A2.1**)	0.2	0.2	0	0	0	0	0.2	0.4	0	0.2	0.2	0	0.2	0	0.2	0	0	0.2
Web site building tools [Dream weaver] (**A2.2**)	0.2	0.4	0	0	0	0	0.4	0	0	0.4	0.4	0	0	0	0.4	0	0	0.4
Web Applications - Ruby on Rails (**A2.3**)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MS Access Database (**A3.1**)	0.6	0.6	0	0	0	0	0	0.6	0	0	0	0	0.6	0	0.6	0	0	0.6
MySQL Database (**A3.2**)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Learning Style																		
Sequential		✓	✓	✓	✓	✓	✓									✓		
Global	✓							✓					✓		✓			✓
Learning Preference		A2.3					A1.1		A2.3	A1.1								

Table 2: Profiles of the non-CS students.

Concept Space	**Students**						
Prior Knowledge	**S19**	**S20**	**S21**	**S22**	**S23**	**S24**	**S25**
Content management systems [CMS] (**A1.1**)	0.2	0.4	0	0	0	0	0.4
Layout design application (**A1.2**)	0.2	0.4	0	0	0	0	0.2
CSS Programming (**A1.3**)	0.6	0.6	0.6	0.6	0.6	0.6	0.6
HTML Programming (**A2.1**)	0.4	0.4	0.4	0.4	0.2	0.4	0.2
Web site building tools[Dream weaver] (**A2.2**)	0.2	0.2	0	0	0	0	0.4
Web Applications - Ruby on Rails (**A2.3**)	0.6	0.6	0.6	0.6	0.6	0.6	0.6
MS Access Database (**A3.1**)	0	0	0.2	0.4	0.2	0.4	0
MySQL Database (**A3.2**)	0	0	0.4	0.2	0.2	0.4	0
Learning Style							
Sequential	✓	✓	✓	✓			
Global					✓	✓	
Learning Preference	A1.1					A2.2	A2.3

Table 3: Profiles of the CS students.

When comparing the second bar of each course aspect against the first one, we note that the core part of A3.1, i.e., MS Access Database, was removed. This indicates that the students were exempted from studying MS Access Database as a result of their prior-knowledge. When comparing the third bar of each course aspect against the second one, we note that the LOAs of the programming based aspects, i.e., A1.3 and A2.3, are significantly reduced. (Note that A3.1 could not be further reduced, as its core part was already trimmed out based on the students' prior-knowledge.) Such change is caused by applying the *non-programming content style filters* at the *aspect group level* of the course. These filters are shown as follows:

Non-Programming Core Filter: {(1, 1), (1, 0.4), (1, 1), (0, 0), (1, 1), (1, 0.2)}
Non-Programming Advanced Filter: {(1, 1), (0, 0), (1, 1), (0, 0), (1, 1), (0, 0)}

where the (r_i, d_i)-pairs in the two filters are used to adjust the LOAs of the aspect groups AG1.1, AG1.2, AG2.1, AG2.2, AG3.1, AG3.2. Applying such content style lets the students focus more on the non-programming based materials while still study the essential parts of the programming based materials to fulfill the course learning outcome requirement.

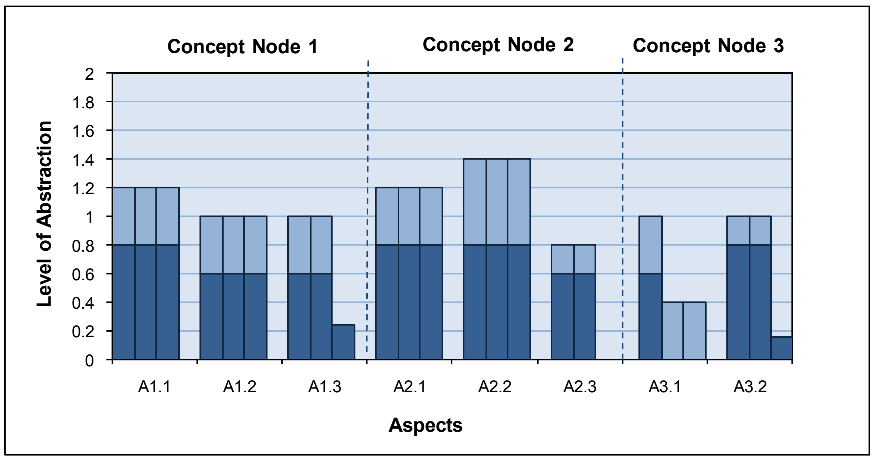


Figure 3: Changes in course dissemination based on prior-knowledge and content style (Group A).

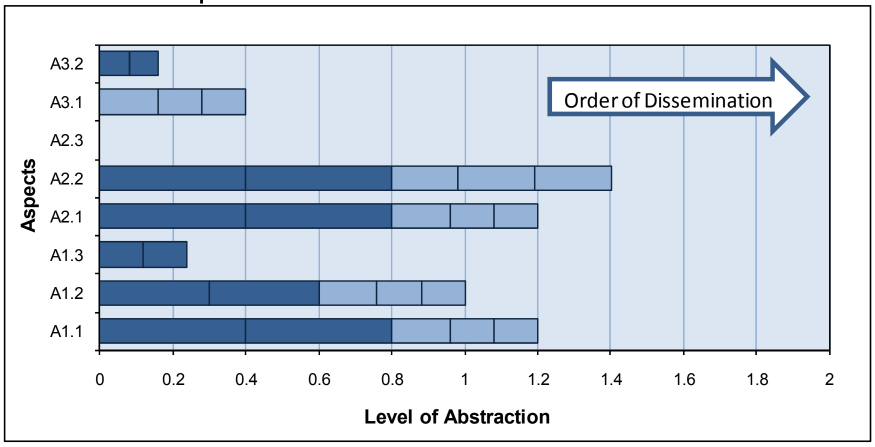
Learning Style: Figure 4 shows the course dissemination after the learning styles have been applied. The course materials are delivered in batches (learning stages) by a prescribed sequence, which is illustrated by following the charts from left to right. As shown in Figure 4(a), to serve students with the *global learning style,* i.e., S1, S8, S13, S15 and S18, each aspect was divided into parts (which are delimited by the black lines in each bar) by applying the global learning style filters. In each learning stage, one part was picked from each course aspect and put together as a whole for dissemination. For the *global learning filters*, five *aspect level* filters were in place and applied one at a time in each learning stage in the following order:

Global Learning Core Filter 1: {(1, 0.5), (1, 0.5), (1, 0.5), (1, 0.5), (1, 0.5), (1, 0.5), (1, 0.5), (1, 0.5)}
Global Learning Core Filter 2: {(1, 1), (1, 1), (1, 1), (1, 1), (1, 1), (1, 1), (1, 1), (1, 1)}
Global Learning Advanced Filter 1: {(1, 0.4), (1, 0.4), (1, 0.4), (1, 0.4), (1, 0.3), (1, 1), (1, 0.4), (1, 1)}
Global Learning Advanced Filter 2: {(1, 0.5), (1, 0.5), (1, 0.5), (1, 0.5), (1, 0.5), (1, 0), (1, 0.5), (0, 0)}
Global Learning Advanced Filter 3: {(1, 1), (1, 1), (1, 1), (1, 1), (1, 1), (1, 0), (1, 1), (0, 0)}

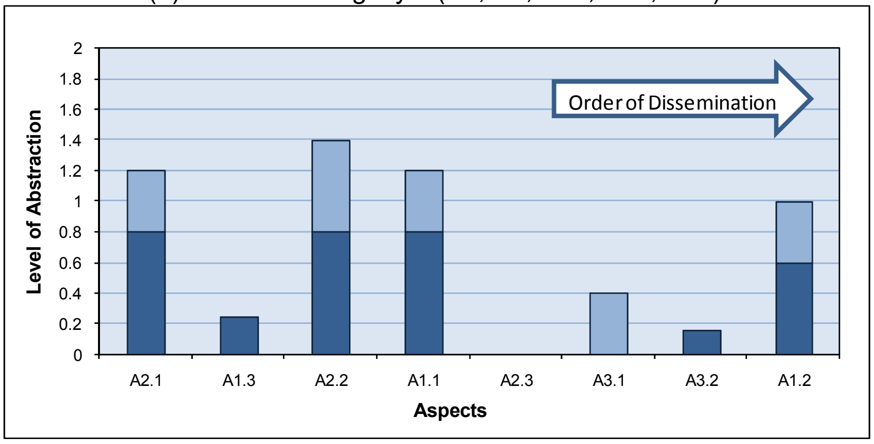
The advanced parts of the course aspects were divided into more parts than the core ones, as the advanced parts of the materials are much difficult to learn. In contrast, to serve students with the *sequential learning style,* i.e., S2, only one course aspect was delivered in each learning stage. Figure 4(b) shows the order of dissemination. For the *sequential learning filters*, eight *aspect level* filters were in place and applied one at a time in each learning stage. Each of the filters is set by having a (1, 1)-pair for the disseminating aspect and a (0, 0)-pair for all other aspects. Since the designed aspect dissemination sequence in our experiment was: A2.1, A1.3, A2.2, A1.1, A2.3, A3.2, A1.2, the first filter was set to:

Sequential Learning Filter 1: {(0, 0), (0, 0), (0, 0), (1, 1), (0, 0), (0, 0), (0, 0), (0, 0)}

where each of the filters was applied to both the core and the advanced parts of each course aspect. Note that the above global and sequential learning filters are also used for the rest of the experiments.



(a) Global learning style (S1, S8, S13, S15, S18)



(b) Sequential learning style (S2)

Figure 4: Course dissemination after applying learning styles (Group A).

CS Students (Group B)

As shown in Table 3, students S19 to S25 are CS students. They have above average knowledge in both CSS Programming (A1.3) and Ruby on Rails (A2.3) before taking the Web Technologies course. The course was tailor-made to allow these students to exempt from studying the core parts of these course aspects. It is shown in Figure 5 by comparing the second bars of A1.3 and A2.3 against the first ones. By considering the students' background, the course was adjusted to allow the students to learn more programming based materials. This is achieved by applying the *programming content style filters* at the *aspect group level* of the course:

Programming Core Filter: {(1, 0.2), (1, 1), (1, 0.2), (1, 1), (0, 0), (1, 1)}
Programming Advanced Filter: {(1, 0.6), (1, 1), (1, 0.6), (1, 1), (1, 0.2), (1, 1)}

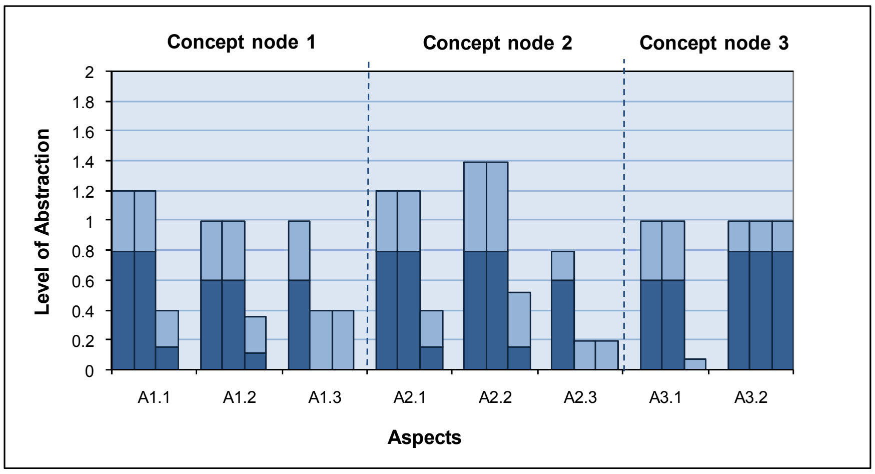
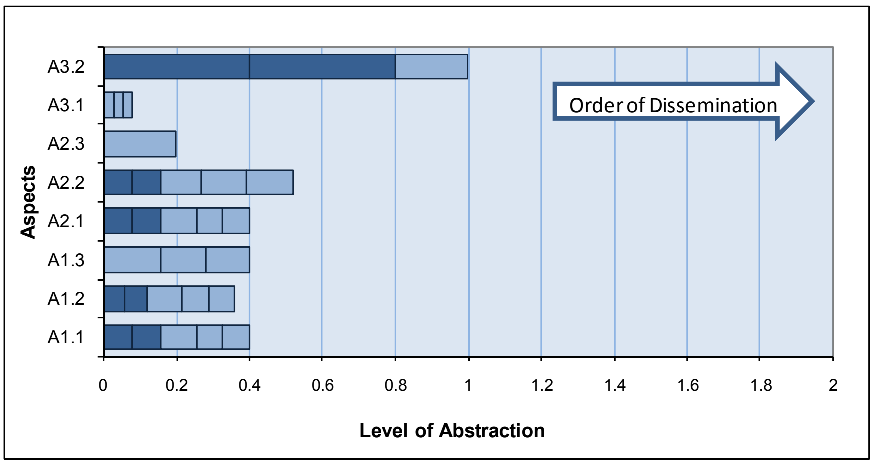
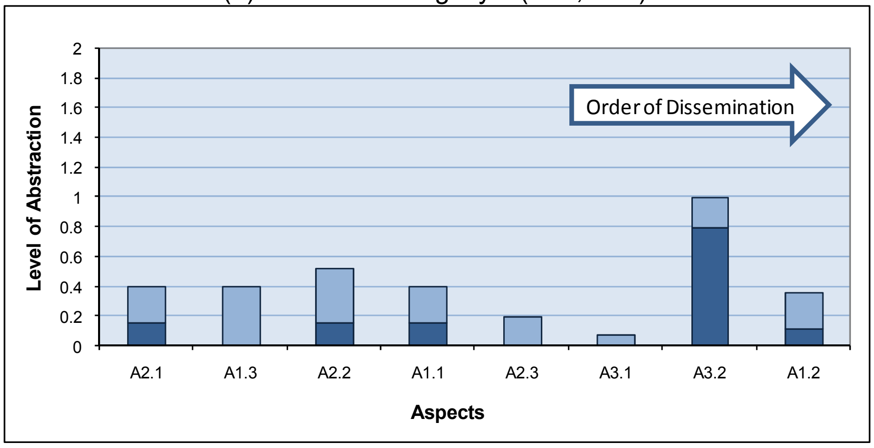


Figure 5: Changes in course dissemination based on prior-knowledge and content style (Group B).

The result of the filtering process is illustrated by the third bar of each course aspect. It shows that the LOAs of the non-programming based materials are reduced significantly. Figures 6(a) and 6(b) show the changes in course dissemination after applying the *global learning filters* and the *sequential learning filters*, respectively.



(a) Global learning style (S23, S24)



(b) Sequential learning style (S19, S20, S21, S22)

Figure 6: Course dissemination after applying learning styles (Group B).

Non-CS Students with Experience in Layout Design Application (Group C)

Students S7, S10 and S11 are non-CS students with above average knowledge in Layout Design Application (A1.2) before taking the Web Technologies course. In addition, student S7 had the sequential learning style, while S10 and S11 did not have any specific learning style. We show the changes in course dissemination in Figures 7 and 8. The experimental results are similar to those in Section 5.1, except that the removal of the core part of the course aspect A3.1 is now replaced by removing the core part of the course aspect A1.2, due to the difference in students' prior-knowledge.

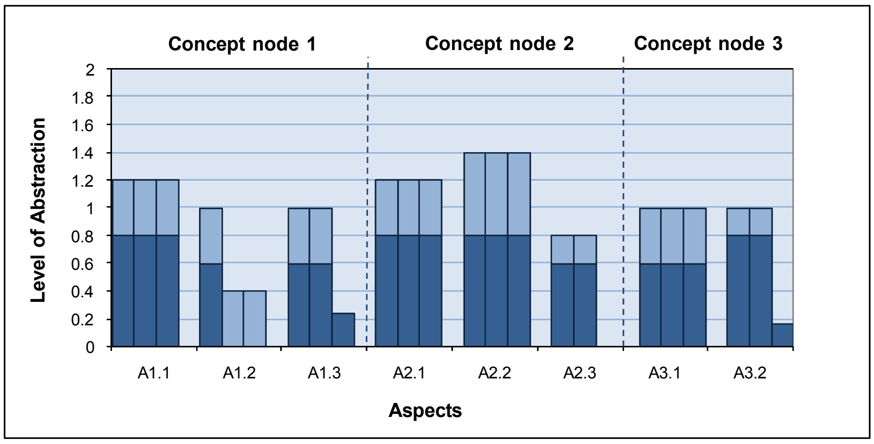


Figure 7: Changes in course dissemination based on prior-knowledge and content style (Group C).

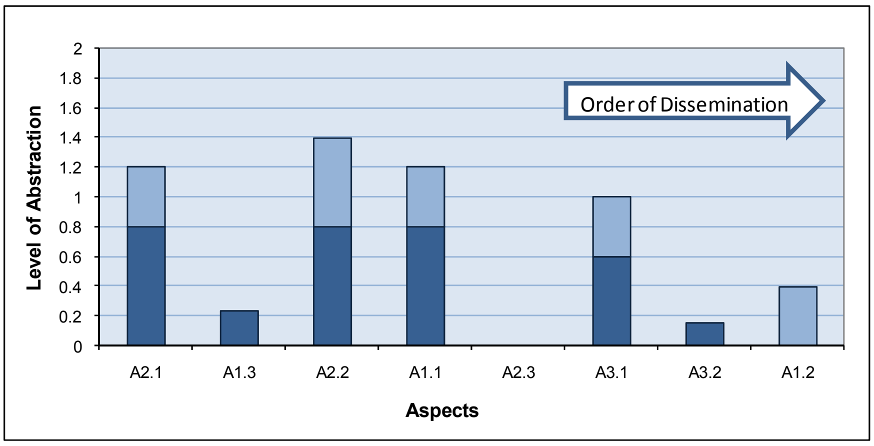


Figure 8: Course dissemination after applying the sequential learning style (S7) (Group C).

Other Non-CS Students (Group D)

The rest of the non-CS students had no knowledge in any course aspects. Hence, by comparing the second bars of all course aspects against the first ones as shown in Figure 9, no change was made to the course design after applying students' prior-knowledge. However, after applying the *non-programming content style filters*, the change in course dissemination was similar to that in Section 5.1, where the LOAs of the programming based course materials were significantly reduced. The tailor-made course dissemination for these students is shown in Figure 10. Interestingly, only students with the sequential learning style and none with the global learning style were in this student group. We believe that this was because these students had never studied any computer related subjects before. Thus, they could not study multiple topics together at the same time.

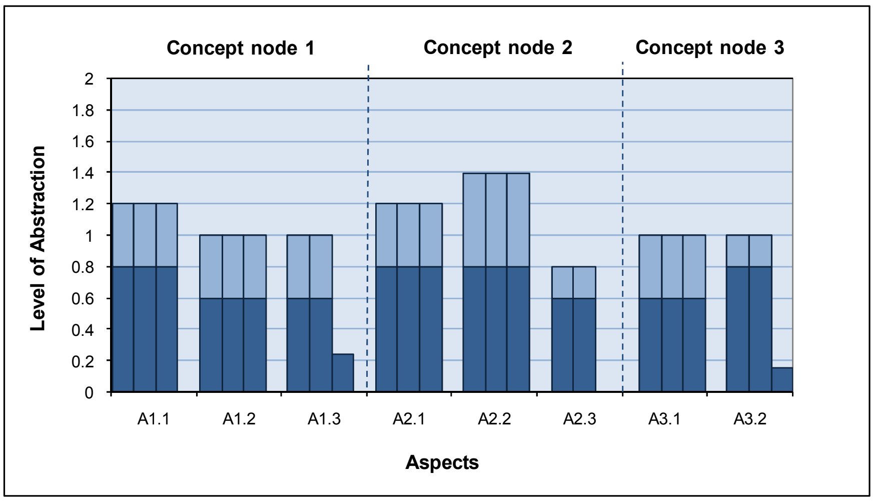


Figure 9: Changes in course dissemination based on prior-knowledge and content style (Group D).

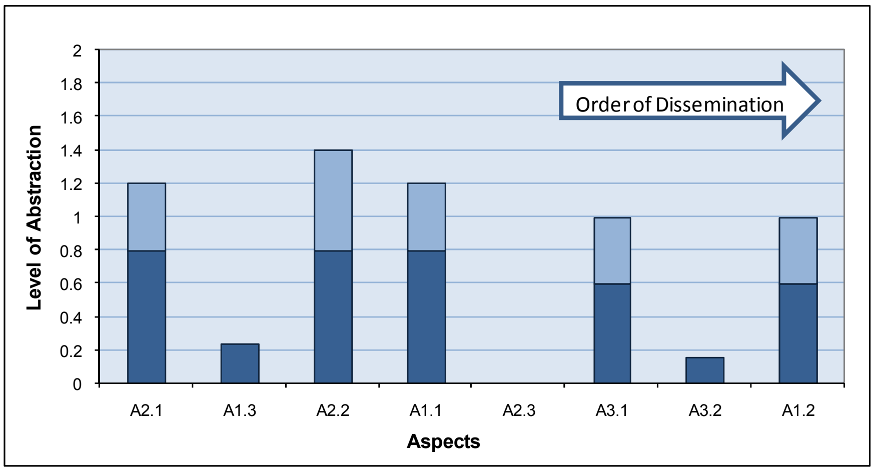


Figure 10: Course dissemination after applying the sequential learning style (S3, S4, S5, S6, S16) (Group D).

Learning Preferences

During our experiments, there were students indicated that they preferred to learn more on certain course aspects, including A1.1, A2.2 and A2.3. Such learning preferences are shown in Tables 2 and 3. To cope with this kind of request, *learning preference filters* were set up as shown in Table 4.

In fact, learning preference filter design is not straightforward and it largely relies on the experience of the course instructor. Here, we show the rationale for the filter settings together with some experimental results. Regarding the learning preference of course aspect A1.1, i.e., Content Management Systems, as it is a non-programming aspect, which should be part of the learning scope of non-CS students, the filter should be designed to process requests from CS students, who are only assigned to study the core part of A1.1. As shown in Table 4, the *preference extraction filter* is set to add the advanced part of A1.1 for CS students to study, while the *course adjustment filter* removes some LOAs from the advanced part of A3.2, which is relatively less important by judging its aspect weight from the course design (Ref. Table 1). As a result, the learning preference of A1.1 from non-CS students, S7 and S10, needed not be processed. To process the learning preference of A1.1 from student S19 (a CS student), the LOAs of the advanced part of A1.1 and A3.2 became 0.4 and 0.1, respectively.

Course Aspect	**Preference Extraction Filter**	**Course Adjustment Filter**
A1.1	Adv.: {(1, 1), (0, 0), (0, 0), (0, 0), (0, 0), (0, 0), (0, 0), (0, 0)}	Adv.: {(0, 0), (1, 1), (1, 1), (1, 1), (1, 1), (1, 1), (1, 1), (1, 0.5)}
A2.2	Adv.: {(0, 0), (0, 0), (0, 0), (0, 0), (1, 1), (0, 0), (0, 0), (0, 0)}	Adv.: {(1, 1), (1, 1), (1, 1), (1, 1), (0, 0), (1, 1), (1, 1), (1, 0.5)}
A2.3	Core: {(0, 0), (0, 0), (0, 0), (0, 0), (0, 0), (1, 0.5), (0, 0), (0, 0)}	Adv.: {(1, 1), (0, 0), (1, 1), (1, 1), (1, 1), (0, 0), (1, 1), (1, 1)}

Table 4: Learning preference filters

The design of the learning preference filters for A2.2 follows quite closely to that of A1.1, as they are of similar nature. Hence, they are also designed to let a CS student study the advanced part of A2.2 while removing some LOAs from the advanced part of A3.2. As student S24 made such a learning preference, the LOAs of the advanced part of A2.2 and A3.2 for S24 became 0.6 and 0.1, respectively.

The learning preference filters for A2.3 serve the need for non-CS students, who are not required to study this course aspect according to the content style. Hence, the filters are designed to allow these students to study certain details of the core part of A2.3 by scarifying the advanced part of the Layout Design Application (A1.2). As a result, the LOA of the core part of A2.3 for student S2 and S9 became 0.3, while the LOA of the advanced part of A1.2 for these two students became 0. In contrast, although student S25 also made this learning preference, this filter did not apply to this student as A2.3 was part of the learning scope.

Discussions

The above results depict the generation of adaptive courses to different types of students. They form a showcase to indicate the *intuitiveness*, *entirety* and *extensibility* of our framework. Here, we discuss these features in detail as follows:

Intuitiveness: Our framework offers a handy way for instructors to develop supports for processing different student characteristics. For instance, to support *content styles*, we use aspect groups to annotate programming and non-programming oriented materials in the course design, followed by setting up appropriate concept filters to define how these types of course materials are disseminated. Likewise, we may also handle *interaction styles* (Papanikolaou et al., 2003), including theory-oriented, exercise-oriented or activity-oriented learning, in a similar way by defining corresponding aspect groups and concept filters.

In contrast, existing methods, such as AES-CS (Triantafillou et al., 2002), InterBook (Brusilovsky et al., 1998), iWeaver (Wolf, 2003) and MOT+AHA! (Stash et al., 2004), adopt the adaptive hypermedia concept (Brusilovsky, 2001) to arrange course materials as a hierarchy of hyperlinked documents, followed by applying adaptive annotation to label the supported learning style(s) to individual pieces of course materials. For course dissemination, either *adaptive navigation* and/or *direct guidance* are used to present selected views of course materials and/or provide suggestions on course material navigation, respectively. In addition, as in MOT+AHA! (Stash et al., 2004), conditional logics may be added to define rules for selecting course materials against the student learning styles and govern the presentation sequences of the course materials. However, this implicitly requires an instructor to have some programming concepts before he/she may be able to construct a course, or alternatively to seek help from a technical assistant. Both are practically unfavorable.

Entirety: Another important feature of our framework is entirety, which is twofold: the *vertical* and the *horizontal* perspectives. In the vertical perspective, our framework offers concept nodes, aspect groups and concept filters. They allow an instructor to process student characteristics comprehensively as a whole to produce better adaptive courses. In particular, our experiments have revealed the processing of students' prior-knowledge, learning styles, content styles and learning preferences along with the results. In the horizontal perspective, our framework allows an instructor to construct a course using concept nodes and to apply concept filters to "shape" the course. This idea is particular important, as the concept filter helps the instructor customize the course by considering all the course aspects as a whole. For instance, when handling the learning preferences in our experiments, we did not only take the learning preferences into account to amend the relevant course aspects, we also adjusted other course aspects to balance the learning workload of the students. To our knowledge, this feature has not been addressed in existing adaptive e-learning systems.

Extensibility: One of the design goals of our framework is to allow an instructor to construct learning materials incrementally and to extend an e-learning course to support different student characteristics. The rationale for us to provide the extensibility feature to our framework is that it is hard for an instructor to get everything in place before offering an adaptive e-learning course, as this may incur a high initial course development time, which may not be practical. In addition, we believe that after getting some on-going teaching experiences and student feedbacks, the ways for processing different student characteristics, such as learning styles, may need to be revised to improve the dissemination of the course materials. Due to the "building block" nature of the concept node and the concept filters, it is easy for an instructor to construct new concept filters to support further learning needs. This is particularly in line with the on-going researches in supporting adaptive e-learning by learning styles (Brown et al., 2009). In contrast, the support of extensibility is either limited or complicated in existing adaptive learning e-learning systems. For example, AES-CS (Triantafillou et al., 2002), InterBook (Brusilovsky et al., 1998), iWeaver (Wolf, 2003) and MANIC (Stern et al., 2000) provide no mechanism to support extensibility. Although MOT+AHA! (Stash et al., 2004) comprises conditional logics that may offer such a feature, as discussed earlier; it is rather difficult for a typical instructor to extend such a system by himself/herself.

CONCLUSION

In this paper, we have proposed a framework based on the concept space and the concept filters. Based on them, we also provide a unified three-tier profiling mechanism, which comprises student, course and resource profiles, to facilitate the adaptive course generation. Our framework uses concept nodes to model course content instead of relying on the construction of complicated ontology hierarchies. The advantage of the concept node idea is that it is much simpler for general instructors to construct their courses. In addition, the framework has an extensible nature, as the concept node structure, aspects and aspect groups, and concept filters can be treated as building blocks and tools to model further learning needs. Hence, they can form the foundation for developing variety of adaptive e-learning systems.

As a future work, we are now investigating the appropriate visual-based user interface design for instructors to construct and manipulate the concept nodes and the concept filters. Our objective is to allow instructors to focus more on the pedagogic issues when designing the concept nodes and the concept filters during the development of an adaptive e-learning course. For example, the instructor may focus on how a course should be disseminated, rather than on defining the numbers in the concept filters.

REFERENCES

Balabanović, M., & Shoham, Y. (1997). Fab: Content-Based, Collaborative Recommendation. *Communications of the ACM*, 40(3), 66-72.

Brown, E., Brailsford, T., Fisher, T., & Moore, A. (2009). Evaluating Learning Style Personalization in Adaptive Systems: Quantitative Methods and Approaches, *IEEE Trans. on Learning Technologies*, 2(1), 10-22.

Brusilovsky, P., Eklund, J., & Schwarz, E. (1998). Web-Based Education for All: A Tool for Development Adaptive Courseware. *Computer Networks and ISDN Systems*, 30(1-7), 291-300.

Brusilovsky, P. (2001). Adaptive Hypermedia, *User Modeling and User Adapted Interaction*, 11, 87-110.

De Bra, P., Aerts, A., Berden, B., de Lange, B., Rousseau, B., Santic, T., Smits, D., & Stash, N. (2003). AHA! The Adaptive Hypermedia Architecture. *Proc. Hypertext and Hypermedia*, 81-84.

Dolog, P., Henze, N., Nejdl, W., & Sintek, M. (2004). Personalization in Distributed e-Learning Environments. *Proc. World Wide Web Alt.*, 170-179.

Felder, R., & Silverman, L. (1988). Learning and Teaching Styles. *Journal of Engineering Education*, 78(7), 674–681.

Freyne, J., Farzan, R., Brusilovsky, P., Smyth, B., & Coyle, M. (2007). Collecting Community Wisdom: Integrating Social Search & Social Navigation. *Proc. Intelligent User Interfaces*, 52-61.

Honey, P., & Mumford A. (1992). The Manual of Learning Styles. *Peter Honey Publications*.

Lehmann, L., Hildebrandt, T., Rensing, C., & Steinmetz, R. (2008). Capture, Management, and Utilization of Lifecycle Information for Learning Resources, *IEEE Trans. on Learning Technologies*, 1(1), 75-87.

Li, Q., Lau, R., Shih, T., & Li, F. (2008). Technology Supports for Distributed and Collaborative Learning over the Internet. *ACM Trans. on Internet Technology*, 8(2), Article No. 5.

Middleton, S., Shadbolt, N., & de Roure, D. (2004). Ontological User Profiling in Recommender Systems. *ACM Trans.on Information Systems*, 22(1), 54-88.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). The PageRank Citation Ranking: Bringing Order to the Web, Technical report, *Stanford Digital Library Technologies Project*.

Papanikolaou, K., Grigoriadou, M., Kornilakis, H., & Magoulas, G. (2003). Personalizing the Interaction in a Web-based Educational Hypermedia System: the case of INSPIRE, *User Modeling and User-Adapted Interaction*, 13(3), 213-267.

Qiu, F., & Cho, J. (2006). Automatic Identification of User Interest for Personalized Search, *Proc. World Wide Web*, 727-736.

Schiaffino, S., Garcia, P., & Amandi, A. (2008) eTeacher: Providing Personalized Assistance to E-Learning Students, *Computers & Education*, 51(4), 1744-1754.

Stash, N., Cristea, A., & De Bra, P. (2004). Authoring of Learning Styles in Adaptive Hypermedia: Problems and Solutions. *Proc. World Wide Web Alt.*, 114-123.

Stern, M., & Woolf, B. (2000). Adaptive Content in an Online Lecture System, *Proc. Adaptive Hypermedia and Adaptive Web-Based Systems*, pp. 227-238.

Studer, R., Benjamins, V. & Fensel, D. (1998). Knowledge Engineering: Principles and Methods. *Data & Knowledge Engineering*, 25(1-2), 161-197.

Triantafillou, E., Pomportsis, A., & Georgiadou, E. (2002). AES-CS: Adaptive Educational System based on Cognitive Styles, *Proc. Adaptive Hypermedia and Adaptive Web-based Systems*, 10-20.

Wolf, C. (2003). iWeaver: Towards 'Learning Style'-based e-Learning in Computer Science Education, *Proc. Australasian Computing Education Conference on Computing Education*, 273-279.

Wu, H., de Kort, E., & De Bra, P. (2001). Design Issues for General-Purpose Adaptive Hypermedia Systems. *Proc. Hypertext and Hypermedia*, 141-150.

Yue, K., & Ding, W. (2004). Design and Evolution of an Undergraduate Course on Web Application Development. *Proc. SIGCSE Innovation and Technology in Computer Science Education*, 22-26.