A Pedagogical Interface for Authoring Adaptive e-Learning Courses

Christopher Watson[1] Frederick W.B. Li[1] Rynson W.H. Lau[2]

[1] School of Engineering and Computing Sciences, University of Durham, United Kingdom

[2] Department of Computer Science, City University of Hong Kong, Hong Kong

ABSTRACT

Due to the differences in background knowledge, learning styles and preferences, individual students may take very different approaches towards learning. In light of this, adaptive educational hypermedia systems (AEHSs) have been developed to offer students personalized learning content to improve their learning outcome. These systems typically use programming scripts and hierarchical course structures as primary techniques to support adaptive e-learning course authoring for different types of students. They require teachers not only to have substantial technical skills, but also to be able to implement theories of learning styles, which are tough requirements. In order for general teachers to participate in authoring adaptive e-learning courses, we have developed a novel and intuitive interface for teachers to graphically visualize and modify the learning contents to be delivered according to different student learning characteristics, based on a set of visual manipulators. The interface also allows teachers to adjust the study load distribution across different parts of a course. We have implemented the proposed interfacing mechanism in an online adaptive e-Learning system and conducted user studies to evaluate its effectiveness.

1. INTRODUCTION

In traditional classroom teaching, all students in a class learn the course in the same way, since learning content has to be delivered to all students at the same time and pace. Adaptive educational hypermedia systems (AEHSs) [4] relax this constraint by constructing personalized learning content for each student according to his/her learning characteristics, such as background knowledge, learning style and preferences. Typical examples of AEHS include AES-CS [14], iWeaver [17] and MOT+AHA! [15]. In general, these systems arrange course materials as a hierarchy of hyperlinked documents, where each document is annotated with its learning style type. When delivering a course, predefined rules and conditions are applied to offer adaptive navigation and/or direct guidance on the learning content. For these rules and conditions to be specified, *programming script* is typically used. An example is the conditional logics offered by MOT+AHA! [15]. In addition, MOT+AHA! also offers a graphical user interface (GUI) to allow relationships among different pieces of learning content to be defined visually. However, it requires the teacher to have substantial technical skill in order to work out the programming scripts and to implement theories of learning styles on their own, which is practically unfavorable. Although GUIs are typically provided, existing systems merely turn information presentation into a well organized and graphical way but cannot convey more pedagogical meaning of an adaptive course design to the teacher.

In this paper, we propose a pedagogical interface for authoring adaptive e-learning courses. This interface provides a graphical illustration on the pedagogical meanings of different course settings, such as how each part of a course should be delivered to a student without certain academic background, and how much content details are distributed across different parts of the course when a particular student learning preference is applied. Such abilities help turn adaptive e-learning authoring into a graph manipulation process, which is easy and intuitive. We organize the rest of this paper as follows. Section 2 briefly summarizes existing works. Section 3 presents our framework on adaptive course authoring and generation. Section 4 describes the design of our pedagogical interface. Section 5 presents the system implementation and shows some experimental results and discussions. Finally, Section 6 briefly concludes this paper.

MTDL'10, October 29, 2010, Firenze, Italy.

2. RELATED WORK

A learning style [7] refers to the concept that different people may learn more effectively in different ways. For example, some people might learn a concept most quickly by performing hands-on exercises or experiments relating to the concept whilst others would learn better by studying the theory behind it. To support different learning styles in an e-learning system [9], where the learning content delivered to different students may be different, a straightforward approach is for the teacher to prepare separate pieces of learning content for different learning styles, and for the e-learning system to select and deliver the appropriate contents to individual students based on their profiles. Most of the existing AEHS, such as AES-CS [14], iWeaver [17] and MOT+AHA! [15], adopt this approach.

In earlier AEHS, such as CS383 [6] and INSPIRE [12], there were no specific user interfaces provided and hence teachers had to spend a substantial amount of effort in developing adaptive e-learning courses. In general, AEHSs require a teacher to annotate the learning content with appropriate learning styles. In addition, a hierarchical structure has to be constructed to link all the learning content together, such that the relevance, the delivery sequences, and the levels of difficulty of individual elements of the learning content can be modeled. To support course authoring, InterBook [2] and MOT+AHA! [15] adopt the markup-based approach, where relevant markup tags can be attached to each piece of learning material to indicate its relevance to particular student characteristics. Such an approach is tedious and prone to user typing mistakes, which are common problems suffered by non-programmer course developers when using InterBook [2]. To support specifying how learning content is actually delivered according to students' learning styles, MOT+AHA! [15] provides programming scripts to teachers to define rules and conditions for learning content delivery. To enhance reusability and exchangeability, HyCo-LD [1] surpasses MOT+AHA! by defining its own programming-liked rules based on the IMS-LD framework [8], which is an educational modeling language. However, even with this improvement, it may still be difficult for general teachers to develop adaptive e-learning course materials by working out programming-like rules directly.

To address this problem, NetCoach [16], MOT+AHA! [15] and HyCo-LD [1] offer graphical user interfaces (GUI), including parameter boxes and option buttons, to simplify the way that teachers define rules and conditions for learning content delivery. These interfaces may also help teachers define relationships visually among different pieces of learning content. Undeniably, the GUIs simplify adaptive e-learning course authoring. It has been predicted that GUIs will become the main approach adopted by future AEHSs to support course authoring [3]. However, at this stage, the GUIs provided by existing AEHSs merely alter how information is presented. Rules, conditions and relationships regarding adaptive learning content are graphically presented, and key elements of learning content are listed out neatly. They cannot convey more pedagogical meaning of adaptive course design to the teachers than these basic relationships.

3. ADAPTIVE COURSE AUTHORING AND GENERATION

In our recent work [10], we proposed a framework based on *concept space* and *concept filters* to support adaptive course authoring where comprehensive student characteristics are considered. The *concept space* represents a confined set of knowledge. It is a data structure for modeling student and course characteristics. The *concept filters* are modifiers that can be applied to a concept space to determine how a course should be delivered. Since our *pedagogical interface* is built on top of this framework, we first summarize the main features of this framework as follows:

Concept Space: An adaptive course is considered as a *concept space* and this space is divided into a number of *concept nodes*. Such a formulation is intuitive as a course typically comprises some major topics, which can be mapped to concept nodes directly.

Concept Node: This is to support a variety of student characteristics, including learning preferences, learning styles, and content styles [10], by customizing the learning content into suitable types of material and levels of detail for individual students. A *concept node*, which is designed to formulate a major course topic, comprises different *aspects*, *aspect groups* and *levels of abstraction*. Two types of concept node are supported, *core* and *advanced* concept nodes, to model the mandatory and the advanced part of a major course topic.

Aspect: A concept node is modeled with a number of *aspects*; each aspect corresponds to a sub-topic under a major course topic. The introduction of aspects is important. They can be used to formulate either related sub-topics under a theme, or alternative presentations of a sub-topic to be taught to different students with different needs. For instance, a teacher can teach the major topic "Webpage Design" with two aspects: *HTML/Javascript programming* and *the use of Webpage authoring packages*. These aspects can be taught as two separate sub-topics, or can be treated as two different approaches for teaching Webpage design to different students, depending on whether a student is studying a computer science related degree or not, for instance.

Aspect Weight: This is an attribute of an aspect, which describes how important the aspect is within a course. An important usage of *aspect weight* is to adjust the study load distribution across different parts of the course.

Aspect Group: An *aspect group* is similar to the idea of *annotation* in existing AEHSs. For example, if an aspect corresponds to the learning content for students who learn more effectively by doing exercises on a concept, we may create an "exercise-oriented" aspect group and assign the aspect to this group. In addition, if this or any other aspects are designed for students with a mathematics background, we may create a "math-student" aspect group to contain them.

Level of Abstraction: The *level of abstraction (LOA)* indicates how much information a topic should be delivered. To our knowledge, only APeLS [5] mentioned LOAs but without detailed elaboration; all other existing AEHSs do not consider LOAs, even though they can easily model LOAs with typical hierarchical structures that they use. Formulating a course topic at different LOAs is useful. For example, they can be used to indicate how detail a topic should be delivered to different types of student. When teaching the topic "Linked list" in C++ programming for a computer science student, we may deliver some in-depth details on this topic, such as physical memory allocation and memory optimization. However, to a mathematics student, for instance, we may only need to illustrate some basic details on this topic, such as how a linked list can be constructed and manipulated, and how it is different from a static data structure, such as an "array". As we have discussed in [10], the idea of LOAs can also help deliver suitable content to global learners [7] and balance study workload across different parts of a course when student learning preferences are considered. In particular, the idea of LOA addresses the deficiency of the concept map used in MOT+AHA! [15] for handling the needs of global learners, since the concept map provides only some low-level relationships among different topics. After reading the concept map, a global learner still needs to go through each topic in detail as a sequential learner does. Pedagogically, it does not offer too much help for global learning, as each topic is still presented separately and with too much detail, which makes it difficult for a global learner to determine the relationships among many topics. However, with LOAs, a global learner can perform global learning by studying some relevant topics at appropriate levels of abstraction.

Concept Filter: Concept node filtering is a mechanism to shape a concept space in order to fulfill the learning needs of 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. Concept filters are critical as they serve as a handy tool to the teacher for authoring adaptive e-learning courses. They support comprehensive student characteristics, without the need to work out any programming scripts. As illustrated in [10], concept node filtering essentially turns the authoring of adaptive e-learning courses into the design of concept filters. By applying the concept filters, a course can be tailored differently for delivering to different students with different learning characteristics. This is somehow similar to processing images in Photoshop, where different filters may change the appearance of an image in different ways.

4. THE PEDAGOGICAL INTERFACE

Advanced AEHSs, such as MOT+AHA! [15] and HyCo-LD [1], resort the delivery of an adaptive e-learning course as a standard search problem, and a teacher needs to write conditional logics or rules to define how each piece of learning content can be matched with different student learning characteristics. This treatment is practically unfavorable as it demands the teacher to have substantial technical skills. More importantly, existing AEHSs do not offer any facilities to convey pedagogical meaning on adaptive e-learning, but require the teacher to understand and implement adaptation rules. As a result, AEHSs are still not widely adopted. The proposed interface design is significantly different from existing AEHSs as it is constructed based on a set of *visual manipulators*, which illustrate the pedagogical meanings of different course settings graphically.

4.1 Design Criteria

Our pedagogical interface aims to help teachers understand and manage how adaptive course content is delivered under different conditions. The interface is developed based on two criteria: complying with the design of concept filters and providing a global picture on how each part of a course contributes to learning the course. The first criterion is essential as our work is based on the idea of concept filters. The second one is a practical need. Example usages include allowing a teacher to incorporate student learning preferences in adaptive courses and adapting a course to students with / without certain academic background.

Complying with the Design of Concept Filters: As mentioned in Section 3, a concept filter is 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. To comply with such a design, we need some form of control for adjusting the values of r_i and d_i. However, such control should not require a teacher to work out the exact values of the (r_i, d_i)-pairs; instead, it should guide the teacher to understand the importance of each piece of concept in a course design and how much detail it will be delivered with. In addition, as some concept filters, namely the global learning and the sequential learning concept filters, are designed to comprise a group of concept filters working one by one for generating learning stages for a course, the interface should also consider how such concept filter groups can be neatly presented.

Providing Global Course View: By complying with the design of concept filters, we can offer a teacher authoring facilities for manipulating the settings on different parts of a course. More than this, in order to help the teacher obtain a high-level understanding of course content delivery, we need to provide some form of presentations showing the contribution of each part of the course, in terms of the importance and the amount of learning content to be delivered, in an adaptive course setting. Meeting this criterion is critical to course design and management, but has not yet been considered in existing AEHSs. Given such a global course view, a teacher can decide which part of the course to adjust in order to match with certain learning preferences or constraints. For instance, it will be easier for the teacher to determine the part of the course to reduce in detail in order to increase the detail of another part that the student is interested in. Another example is that a teacher can treat a global course view as a tool to decide how the study load distribution across different parts of a course should be adjusted based on some teaching feedback. This design criterion pushes our work forward to make AEHSs more practical.

4.2 Visual Manipulators

The pedagogical interface comprises three types of visual manipulators for authoring adaptive e-learning courses. They are the *standalone*, *group* and *sequence* manipulators, and are implemented using JavaScript to ensure cross-browser compatibility. Specifically, we use the Raphaël JavaScript (SVG) library [13] to support interactive graph drawing and manipulation.

Standalone Manipulator: The standalone manipulator is designed to generate concept filters for manipulating the settings of a concept space. Figure 1 shows a screen shot of a standalone manipulator in our implementation. The manipulator is divided into the upper (blue) and the lower (red) sections, which handle the settings of core and advanced concept nodes within the concept space, respectively. Within each section of the manipulator, a curve operator and a set of bar operators are provided for adjusting the LOAs and the aspect weights, respectively, for all aspects of each concept node within the concept space. In addition, the black vertical lines partitions the manipulator into zones, where each zone corresponds to a core and advanced concept node pair with their aspects shown in separate columns.

To manipulate the LOA of an aspect, a teacher is required to drag the corresponding circle point on a curve operator vertically to the desired level. The Y-axis at the left-hand side of the manipulator gives the scale of the LOA values. Aspect weights are manipulated in a similar way, where a teacher is required to drag the circle point on top of the corresponding bar operator vertically to the desired level. The Y-axis at the right-hand side of the manipulator gives the scale of the aspect weight values. For the instructor to determine the most appropriate LOA of an aspect, we have incorporated a "Preview Mode" into the manipulator. This uses jQuery Modal dialog forms to provide the teacher with a preview of the learning content that a student will receive based upon the current LOA setting. Figure 2 shows two examples of the learning content preview mode depicting the effects of increased and reduced LOA on an aspect.

The design of the curve operator and the bar operators complies the criterion of offering a teacher a global course view. The bar operators let the teacher visually understand the importance of different aspects in the course. The curve operator uses a curve connecting all aspects at their LOAs. This intuitively illustrates a comparison of the amount of details delivered by different aspects of the course. Based on these two types of operators, the teacher can easily visualize the emphasis of a course and adjust it accordingly. The standalone manipulator is suitable for generating concept filters to handle *course setup*, *student learning preferences* and *teacher's teaching preferences*.

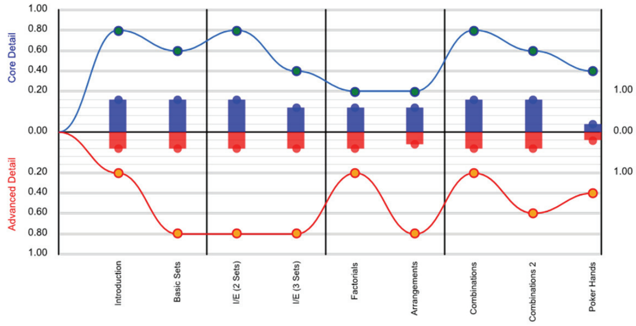
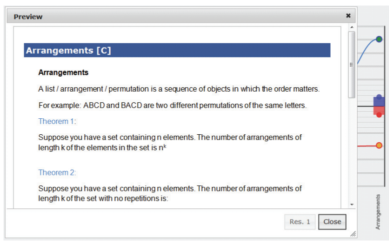
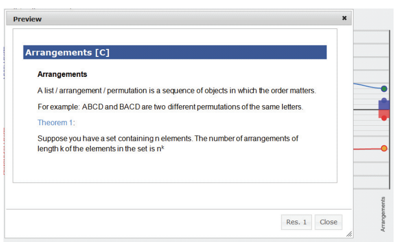


Figure 1. Screen shot of an individual manipulator.



(a)



(b)

Figure 2. Examples of learning content preview mode: (a) increased LOA, and (b) reduced LOA.

Group Manipulator: The group manipulator is designed for adjusting the setting of aspect groups. Figure 3 shows a screen shot of this type of manipulator. The design of the group manipulator is similar to that of the standalone manipulator. As the focus here is on the setting of aspect groups rather than individual aspects, the bar operators are not included. Instead, additional slider operators are attached to the curve operators. As shown in Figure 3, there are two aspect groups set up. They are the "Theoretical" and the "Practical" groups. Four slider operator sets are shown in rows sequentially for controlling settings of the "Theoretical" and the "Practical" aspect groups in the core concept nodes as well as in the advanced concept nodes, respectively. Specifically, a slider operator set is designed to control the LOA settings of all aspects in a course that belong to a certain aspect group, where an independent slider is assigned to each concept node to provide a fine-control of LOA settings. For example, the slider operator set showed in the first row controls the LOA settings of all aspects belonging to the "Theoretical" aspect group in the core concept node. On the other hand, a group manipulator contains two sets of curve operators. The set of darker colors (blue and red) indicates the current LOA settings, and the set of lighter colors (green and orange) indicates the LOA settings before any change. Figure 3 shows the effect of reducing the emphasis on the "Theoretical" content of the course by 50%, which is done by moving the sliders at the first and the third rows. The group manipulator is suitable for generating concept filters to handle *content style* and *learning styles* that are based on different types of teaching materials, such as text, auditory and visual styles.

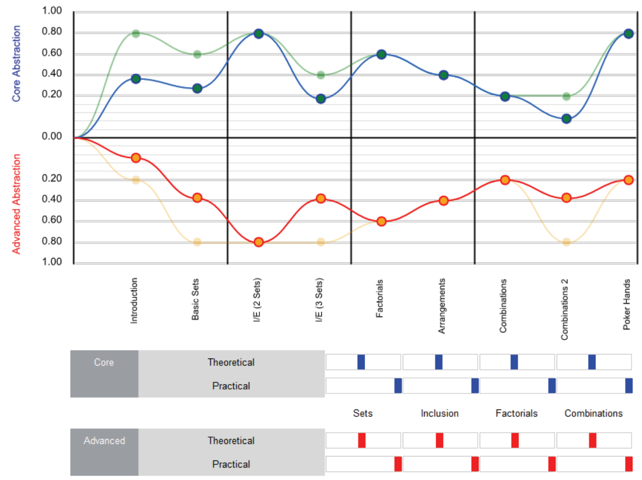
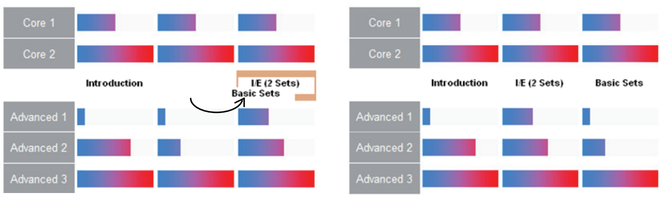


Figure 3. Screen shot of a group manipulator.

Sequence Manipulator: The sequence manipulator is designed to produce a group of concept filters for creating learning stages. It is suitable for generating concept filters to handle *global learning* and *sequential learning*. As shown in Figure 4, the layout of this manipulator is similar to that of the slider operators in the group manipulator. The sequence manipulator comprises two parts, a *stage table* and a *component table*. The stage table consists of a number of rows, some representing core learning stages and the others representing advanced learning stages. We display the core learning stages one by one on the upper section of the table, followed by the aspects and advanced learning stages. Our system lets the teacher modify the number of core or advanced learning stages, which are used to fine-tune the partitioning of learning content, in terms of LOA, to be delivered over a number of stages. The component table is a LOA palette, where each block corresponds to a percentage of content details (in terms of LOA) to be delivered and the values available are in increments of 10%. The more color filled in the block, the larger percentage of content details will be delivered in a learning stage. This table is provided so that a teacher may easily define the LOA of a particular aspect of a learning stage by dragging and dropping a suitable block from the component table into a cell in the stage table corresponding to the desired aspect and learning stage. We do not allow a teacher to specify an arbitrary LOA in the interest of simplicity. With the block approach to present LOAs, a teacher can easily visualize the building up of learning content over time by comparing the amount of color in subsequent learning stages. However, for clarity, the exact percentage that a block represents is displayed on mouse hover. On the other hand, as some learning styles may require a different aspect order, we let the teacher manipulate the horizontal sequencing of learning content by dragging and dropping aspect columns. Figure 5 shows that the second aspect column, "Basic Sets," has been dragged and dropped to swap with the third aspect column, "I/E (2 Sets)," to delay its order of delivery. Note that the row and the column sequences of the stage table are designed for generating concept filters to handle *global* and *sequential learning styles*, respectively.

Figure 4. Screen shot of a sequence manipulator.



(a) (b)

Figure 5. Screen shots of dragging and dropping an aspect column in a stage table and the result.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

We have implemented an adaptive e-learning system based on the framework presented in [10] and used our pedagogical interface proposed here to support adaptive course authoring. The system implementation is done by extending Moodle [11], which is a popular open source e-learning system already used by a number of institutions. Our primary reason for using Moodle is to reuse its base functionality, such as course construction and user accounts. In addition, incorporating our framework into an existing e-learning system also has the advantage of providing a familiar interface for teachers with previous experience of using Moodle or systems alike. Furthermore, Moodle includes a wide range of learning tools such as discussion forums, chat facilities, and interactive quizzes, which we have incorporated into the implementation for use as tools to reinforce the students learning process. Using this system implementation, we have constructed a sample course on Discrete Mathematics consisting of nine topics. To structure the course, we have created four concept nodes and used two aspect groups to describe the content styles of the material. Due to page limitation, we do not list out the details of the course setting. To confirm our contribution, we adopt usability testing as the evaluation methodology; it is to collect and analyze user experience in using our system. This well matches the emphasis of our work, which is on user understanding. We have conducted evaluations on two types of users, namely *teachers* and *students*, who use our system to author adaptive e-learning courses and to study e-learning courses, respectively. However, since the focus of our work is on course authoring rather than student learning, we elaborate the evaluation on teachers in more detail while only briefly discuss the evaluation on students.

Evaluation on Course Authoring: The research question which we were investigating was 'Can our visual manipulators and concept filtering approach aid ordinary / non-technical teachers to create adaptive courseware?' To assess this, we conducted an experiment using a sample of eight teachers from a local special needs school. We were interested in this school in particular, as their teachers are used to adapting their courses on a daily basis within the classroom environment to make them more suitable to students who possess a range of specialist learning needs. The majority of teachers who took part in this experiment possessed extensive teaching experiences: 20+ years (5 teachers), 5-9 years (1 teacher), and 0-4 years (2 teachers). 6 teachers had never used an e-learning system before. 2 teachers possessed a prior programming background. One teacher possessed prior experience of using Pascal, but cited that this was "very basic and a long time ago!" The second teacher possessed prior experience of XHTML, XML, JavaScript and ActionScript. As these teachers were still ordinary classroom teachers, who did not program on a regular basis, we included them in the sample.

The teachers were invited to attend a workshop, which lasted for approximately 45 minutes. Teachers were first introduced to general e-learning systems followed by a brief talk on major aspects of our system, including the concept filtering mechanism and the idea of level of abstraction of course materials. After viewing a demonstration of and interacting with each manipulator, teachers were asked to answer a number of questions related to the interaction, clarity and the manipulators. At the end of the session, the teachers were provided with a discussion and demonstration of the authoring techniques used in InterBook [2] and MOT+AHA! [15], followed by a survey questionnaire. Each question in the survey had five possible responses, where "5" indicated a strong positive response to a statement and "1" indicated a strong negative response. After collecting the questionnaires, we averaged the total points for each question and rounded to the nearest response to obtain the average group response.

The feedback provided on the standalone manipulator was mainly positive. Teachers indicated that the detail of an aspect core or advanced material could be identified quickly by using the curve operators (78%), and that the default ordering could be identified fairly quickly by using the bar operators (70%). Identifying the overall LOA by using distances among the points was slightly lower (68%) with a group response indicating that the overall abstraction of an aspect could be identified but not quickly. Ease of interaction scored very highly (93%) indicating that the teachers found it very easy to adjust the LOA of individual aspects by dragging the points on the curve operator vertically. Most of the teachers found the preview mode to be a very useful feature (70%), indicating that a teacher required this feature to work out where to place the points on a curve operator and would find it difficult to set the LOA without it. The group manipulator was again met with a positive response, and teachers indicated that the uniform LOA adjustment for the entire aspect group was useful (77%). However, one teacher pointed out that providing individual adjustment to certain aspects would be beneficial.

The final manipulator which the teachers tested was the sequence manipulator. Ease of interaction scored very highly in terms of both populating learning stages with component blocks (93%), and reordering aspects via drag and drop interaction (95%). Teachers indicated that it was easier to identify the aspect ordering from this interface (78%) rather than using the bars in the standalone manipulator. Whilst the teachers indicated that they could clearly visualize the building up of material over a number of learning stages (73%), they could not decide (65%) whether the amount of material which a block represents could be identified visually. It was suggested that the detail between large and small blocks (e.g., 100% and 50%) could be identified easily; however, distinguishing a sequence of blocks such as 60%, 70%, 80% was difficult. Adding a label to the blocks was suggested as a way of showing the amount of material they represent. Nevertheless, on the whole, the teachers commented that the "blocks were clear and flexible".

When the teachers were asked if they could use our manipulators to construct an adaptive course, they indicated an agreement response for all manipulators with scores of 80% for the standalone manipulator, 71% for the group manipulator, and 70% for the sequence manipulator. To compare our approach to existing work, the teachers were provided with a demonstration and discussion of the authoring techniques using InterBook [2] and MOT+AHA! [11] and asked to comment on whether or not they would be able to create adaptive courseware using these systems. The response was split for InterBook (58%), with teachers commenting that whilst they could use the annotation format, they would find it "tedious to do so". The group disagreed that they would be able to construct a course using AHA! (48%), agreed that on the whole the filtering approach that our system offered was simpler than both existing solutions (73%) and that they could construct an adaptive course using the system (78%).

The evaluation of the questionnaires suggest that our visual manipulators and concept filtering mechanism have provided ordinary or non-technical teachers with a tool to construct adaptive courses whilst taking into account a variety of learning needs and preferences. All manipulators were met with positive responses from the teachers, and after a 45 minute workshop, all teachers indicated that they would be able to create an adaptive course by using the manipulators without requiring further technical assistance. Considering that the majority of the group also possessed no prior experiences of e-learning systems and that after a short training, they could construct an adaptive course and implement complex learning strategies by using our system, this demonstrates that our pedagogical interface provides a both intuitive and effective solution to the problem of authoring adaptive e-learning courses.

Evaluation on Student Learning: We have recruited 61 students from various departments of Durham University to take part in the study, where 40 students were either Mathematics undergraduates or had already achieved grade B or above in A-Level Mathematics, while the rest of the students do not have an A-Level or above Mathematics background. We tested whether the adaptive e-learning content delivered can help improve student performance in their learning. The result obtained confirms the initial hypothesis that students who received learning content which was tailor-made to their background learn more than their peers who received material which was mismatched to their background. The students with Mathematics background who received learning content matched to their learning styles scored on average 6% higher on the post-course quiz than their peers who received mismatched adaptive learning content, while the matched Non-Mathematics students scored on average 5% higher than their mismatched peers.

6. CONCLUSION

This paper presents a novel pedagogical interface to support adaptive e-learning course authoring. This interface provides a graphical illustration on the pedagogical meanings of different course settings, which can both enhance the teacher's understanding on adaptive e-learning and can turn adaptive e-learning authoring into a graph manipulation process, which is easy and intuitive. We have implemented a workable system and have conducted a workshop with some teachers in a local special needs school to study the usability of our system and have collected their opinions between using our system and two reputable existing systems. Results show that our pedagogical interface is a favorable solution for supporting adaptive e-learning course authoring, particularly emphasizing its understandability and learnability. At this stage, we have only tested and implemented our framework with the learning styles described in this paper. As a future work, we are now investigating the incorporation of other types of learning styles into our framework.

7. REFERENCES

[1] A. Berlanga and F. García-Peñalvo, "Learning Design in Adaptive Educational Hypermedia Systems," *Journal of Universal Computer Science*, **14**(22):3627-3647, 2008.

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

[3] P. Brusilovsky, "Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools", *Authoring Tools for Advanced Technology Learning Environment*, Kluwer Academic Publishers, pp. 377-409, 2003.

[4] P. Brusilovsky and C. Peylo, "Adaptive and Intelligent Web-based Educational Systems," *International Journal of Artificial Intelligence in Education*, **13**(2-4):156–172, Apr. 2003.

[5] J. Canavan, *Personalized e-Learning through Learning Style Aware Adaptive Systems*, Published dissertation, University of Dublin, 2004.

[6] C. Carver, R. Howard, and W. Lane, "Enhancing Student Learning Through Hypermedia Courseware and Incorporation of Student Learning Styles", *IEEE Trans. on Education*, **42**(2):33-38, Feb. 1999.

[7] R. Felder and L. Silverman, "Learning and Teaching Styles in Engineering Education," *Journal of Engineering Education*, **78**(7):674–681, 1988.

[8] IMS-LD, URL: http://www.imsglobal.org/learningdesign/

[9] Q. Li, R. Lau, T. Shih, and F. Li, "Technology Supports for Distributed and Collaborative Learning over the Internet," *ACM Trans. on Internet Technology*, **8**(2):Article 5, Feb. 2008.

[10] F. Li, R. Lau, and P. Dharmendran, "A Three-Tier Profiling Framework for Adaptive e-Learning," *Proc. ICWL*, pp. 235-244, Aug. 2009.

[11] Moodle, URL: http://moodle.org/

[12] K. Papanikolaou, M. Grigoriadou, H. Kornilakis, and G. Magoulas, "INSPIRE: an INtelligent System for Personalized Instruction in a Remote Environment", *Proc. Adaptive Hypertext and Hypermedia*, pp. 13-24, Jul. 2001.

[13] The Raphaël JavaScript (SVG) library, URL: http://raphaeljs.com/

[14] E. Triantafillou, A. Pomportsis, and E. Georgiadou, "AES-CS: Adaptive Educational System based on Cognitive Styles," *Proc. Adaptive Hypermedia and Adaptive Web-based Systems*, pp. 10-20, May 2002.

[15] N. Stash, A. Cristea, and P. de Bra, "Authoring of Learning Styles in Adaptive Hypermedia: Problems and Solutions," *Proc. World Wide Web Alt.*, pp. 114-123, May 2004.

[16] G. Weber, H. Kuhl, and S. Weibelzahl, "Developing Adaptive Internet Based Courses with the Authoring System NetCoach", *Proc. of Adaptive Hypertext and Hpermedia*, pp. 35-48, Jul. 2001.

[17] C. Wolf, "iWeaver: Towards 'learning style'-based e-Learning in Computer Science Education," *Proc. Australasian Conf. on Computing Education*, pp. 273-279, Feb. 2003.