Università degli Studi di Roma Tre
Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79 – 00146 Roma, Italy.

Design of a Virtual Environment for Comparing Curriculum Sequencing Algorithms

Carla Limongelli[†], Stefano Pirone[†], Filippo Sciarrone[†], Giulia Vaste[†]

RT-DIA-139-2008 **Dicembre 2008**

† Università di Roma Tre,
Via della Vasca Navale, 79
00146 Roma, Italy
e-mail: `limongel@dia.uniroma3.it, stefano.pirone@gmail.com,`
`f.sciarrone@dia.uniroma3.it, vaste@dia.uniroma3.it`

ABSTRACT

Curriculum Sequencing is one of the most appealing challenges for educational environments where learners need adapted courses, i.e., courses tailored to their own personal traits. In this work we address the problem of how to compare and test different Curriculum Sequencing algorithms in order to reason about them through a self-contained and homogeneous environment. We propose LS-LAB, a framework for comparing and testing different Curriculum Sequencing algorithms in learning environments such as Educational Hypermedia. LS-LAB has been designed to run different Curriculum Sequencing algorithms, each of them provided with its own Student Model representation. In this framework, the Knowledge Domain must be IEEE LOM compliant, while, through a suitable GUI, one can insert new algorithms or run already available ones. We are carrying out the implementation by using a 3-tier Java application technology, in order to provide this environment available on the Internet. Finally we perform an application example.

Contents

1 Introduction

The rapid growth of the Internet is enabling a more widespread use of distance learning based on Web-oriented systems such as Web-based Educational Hypermedia (WBEH) and Learning Management Systems (LMS). In this context, the pedagogical strategy behind a course is crucial, such as the capability of a system to automatically tailor the course to the needs and interests of each individual student. In fact Personalization and Adaptation, as opposed to the traditional *one-size-fits-all* approach, are more and more sought in educational systems [4]. *Curriculum Sequencing* is one of the most interesting challenges in educational research area: research in this field aims to automatically produce a personalized sequence of didactic materials or activities, on the basis of each student's needs and interests, by dynamically selecting the most appropriate didactic materials at any moment [5]. Several approaches and AI techniques addressing this issue have been proposed in the literature: rule-based sequencing, as in the AHA! system [3]; planning-based sequencing, as in the LS-PLAN system [8] and in the work of Baldoni et al. [1]; graph-based sequencing as in the IWT system [11], KBS-Hyperbook system [7], and Lecomps system [12].

In this paper we present a framework called LS-LAB, at its early stage of development, with the aim to compare different Curriculum Sequencing algorithms. The motivation behind this effort arises from the fact that, while from one hand the number of proposals in this field are increasing, on the other hand, presently, there is the lack of an environment where one can actually test and compare different sequencing algorithms. In fact to really compare them, it needs to start from the same conditions and run on the same learning material. Moreover, our system aims to provide researches with an almost *ready-to-use* environment, allowing for *low-cost* experimentations.

In the literature there are several proposals of frameworks designed for evaluating, testing and comparing different kinds of systems and algorithms. For example in [15] a framework for evaluating the performance of user modelling servers is presented. Weibelzahl and Lauer presented a framework to evaluate different adaptive Case-Based Reasoning systems [13], while in the work of Baldoni et al. a framework for comparing Adaptive Educational Hypermedia is presented [2].

In LS-LAB system several adaptive components of an adaptive educational environment are involved, which, through suitable software interfaces , e.g. parsers, must run in the same environment.

This paper is structured as follows: in Section 2 the LS-LAB system, with its components and procedures is shown. In Section 3 is reported the first experimentation, based on two sequencing algorithms: the first one based on a topological sorting vs. the second one based on automated planning. In Section 4 conclusions are drawn.

2 The LS-LAB System

The main issue concerning the design of such an environment is the heterogeneity of the different sequencing techniques, each of them with its own particular *SM* representation, together with their different DK that have to be uniformly parsed and represented in order to have a homogeneous environment. Other problems concern the building of all those software interfaces among algorithms and the technical environment, needed to correctly run them.

2.1 Background

In the following we give some definitions about some components used by the system.

- *Knowledge Item* (KI). A KI is an atomic element of knowledge about a given learning topic.

- *Knowledge Domain* (KD). It is the set of all the KIs related to a particular course: $KD = \{KI_1, KI_2, \ldots, KI_n\}$.

- *Cognitive State* (CS). The CS is the set of all the KI possessed by the student with respect to a given topic: $CS \subseteq DK$.

- *Learning Style* (LS). A LS is a 4-tuple:

$$LS = \langle D_1, D_2, D_3, D_4 \rangle, \quad with\ D_i \in [-11, \ldots, +11], \quad i = 1, \ldots, 4$$

  being each D_i a Felder-Silverman Learning Style Dimension [6], i.e., D_1: active-reflective, D_2: sensing-intuitive, D_3: visual-verbal, D_4: sequential-global.

- *Course.* A Course is a particular set of Learning Nodes (LN), created by the teacher about a particular topic.

- *Target Knowledge* (TK). The TK for a given course is a subset of KIs that represent the knowledge that has to be acquired after having taken the course.

- *Learning Node* (LN). A LN is a 5-tuple:

$$LN = \langle LM, AK, RK, LS, T \rangle \ \text{ where}$$

  LM is the Learning Material, i.e., any instructional digital resource.

  AK Acquired Knowledge. It is a KI that represents the concept that has to be acquired after having studied the contents related to a given LN.

  RK Required Knowledge. It is the set of KI necessary for studying the material of the node, i.e., the cognitive prerequisites required by the AK associated to the node.

  LS Learning Styles.

  - *Learning Object Sequence* (LOS). It is the sequence of LN selected by a given sequencing algorithm.

- *Super Student Model* (SSM). The SSM is a set of data given by the union of all the students characteristics represented in the different SMs related to the available sequencing algorithms. We distinguish two main parts:

$$SSM = \{Name, Age, LS, \ldots\} \cup \{Context, CS, CourseLevel \ldots\}$$

  The first set contains information independent on the specific course, while the second one contains information that are specific for a given course. It dynamically increases, as new algorithms, and related student models, are inserted into the system.

We assume that each LN in a course is IEEE-LOM compliant. To this purpose we show how the LN data given in the definitions above are mapped into some suitable LOM metadata. For example the contents of the fields $<RK>$ e $<AK>$ are represented by means of the tag <purpose> of the last IEEE-LOM category, <classification>; each element of the <taxonPath>, <taxon> is composed by an identificative <id> and by a string <string>, that represents the name of a prerequisite, or an acquired knowledge.

In any case, researchers that want to compare their sequencing algorithms to other ones in LS-LAB, have to build LNs compliant to the IEEE-LOM specification and they have to provide, besides the executable files or source code, a set of parsers for data translations.

2.2 The System Architecture

In this Subsection we show the LS-LAB system in all its logical and functional components, starting from the Fig. 1 where dashed arrows represent the input given by the researcher. The system is composed of the following functional modules:

- *GUI*. It is the graphical environment shown in Fig.2, composed of five components. Through the *insert your data* component the researcher can input the personal data of the student, i.e., name, surname, age together with her Learning Styles (LS). Through the *Course Information* component one can select the course of interest, while through the *Starting Knowledge Selection* and the *Target Knowledge Selection* components the researcher can input respectively the Student Starting Knowledge and the Course Target Knowledge. Finally, through the *Algorithm Selection* component the algorithm currently present in the environment can be selected.

- *Student Model Generator* (SMG). This module generates the particular SM_i related to the particular algorithm A_i to be run. Firstly, the SMG takes in input the researcher's SM selection coming from the GUI, among all the SMs contained in the current *Super Student Model* (SSM). Secondly, it runs the $Parser_i$ producing as output the SM_i, i.e., that particular SM representation needed by the algorithm A_i.

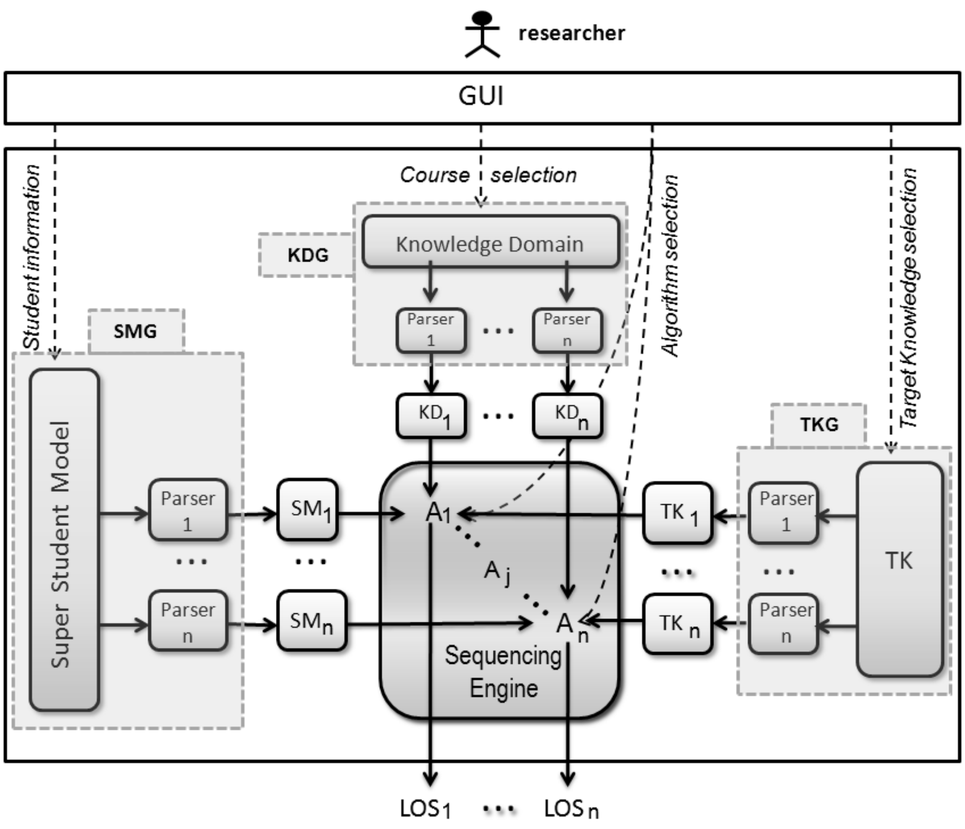
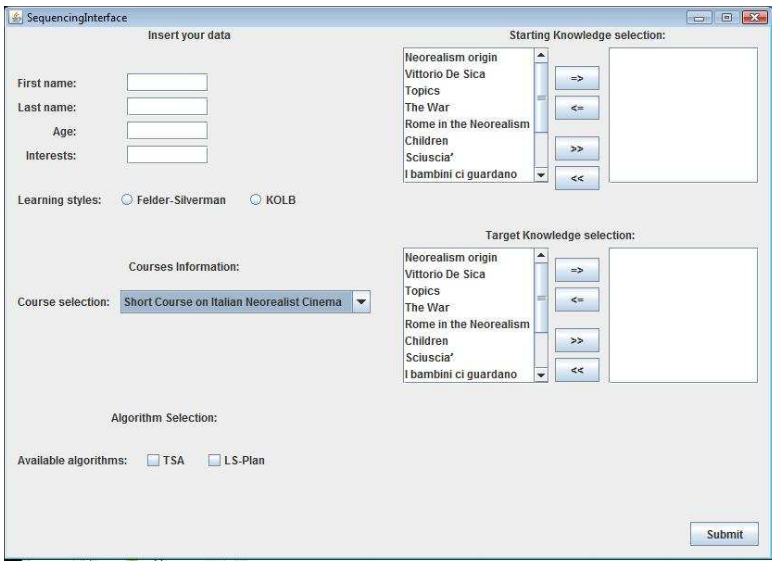


Figure 1: The Functional Schema of the LS-LAB System.

- *Knowledge Domain Generator* (KDG). This module takes as input the $Course_i$ selected by the researcher giving as output the KD_i related to that course, by launching the right parser in order to obtain the right KD representation adapted to be managed by the algorithm A_i.

- *Target Knowledge Generator* (TKG). This module takes as input the TK selected by the researcher, and consequently, through a suitable parser, produces as output the right TK_i representation ready to be managed by the A_i algorithm.

- *Sequencing Engine* (SE). This module is the core of the system. After having taken as input the sequencing algorithm selection, for example A_i, among all those algorithms currently present in the system, it runs A_i on the 3-ple $I =< TK_i, KD_i, SM_i >$.



Course selection: Short Course on Italian Neorealist Cinema

Figure 2: LS-lab GUI.

2.3 The System Workflow

LS-lab can be used in two different scenarios. In the first one a researcher can run two or more algorithms currently present in the system, while in the second one she can insert a new algorithm in the system comparing it with other available algorithms. The insertion of a new algorithm needs the effort to uniform its associated SM, its KD, and its TK representations to the LS-lab standard. This effort means that we need to provide information about the algorithm input data, to provide the executable file of the algorithm, and to develop the necessary parsers and adapters for SM, KD, and TK. So a researcher, who wants to insert her algorithm into the LS-lab system, has to provide the LS-lab developers with:

- the executable file of the algorithm;

- the description of the algorithm input data;

- an *xml* file, describing the SM: it will be used by LS-lab developers to extend the SSM and to provide a suitable adapter between the SSM and the actual SM, that can be given as input to the algorithm;

- a domain description compliant to IEEE-LOM standard and, if required, the semantics of the used tags: it will be used by LS-lab developers to provide a suitable parser, that will translate the IEEE-LOM metadata of the learning materials of a given domain into the correct input for that particular sequencing algorithm;

- a TK description to allow LS-LAB developers for providing a suitable parser to give the correct input to the algorithm;

A researcher who wants to run two or more algorithms already present in the system, has to fill-in the GUI of the system, giving the information about a simulated student, about the domain to be used and about the TK. Moreover she has to select the algorithms she wants to compare. The SM, DK and TK information are consequently translated by the parsers associated to the selected algorithms and sent as inputs to the algorithms themselves. Their outputs are then shown to the researcher.

3 LS-LAB at Work

In this Section we present an example of the use of the LS-LAB system for comparing two sequencing algorithms. The first algorithm is the one used by the LS-PLAN system [8, 9], based on the Pdk (Planning with Domain Knowledge) planner [10]. Pdk conforms to the *planning as satisfiability* paradigm, and the logic used to encode planning problems is the propositional Linear Time Logic [14]. The second algorithm is a classical Topological Sort Algorithm (TSA), used in the literature by a number of adaptive educational systems such as the IWT system [11], the KBS-Hyperbook system [7], and the Lecomps system [12]. The TSA uses student's previous knowledge for building a personalized sequence performing a depth first traversal, while LS-PLAN uses both student's previous knowledge and LS for sequencing. The KD used in this experiment is the *Italian Neorealist Cinema*, as shown in Fig. 3, where the prerequisite relations among concepts are represented through arcs. Some of these concepts are linked to more than one learning material. In particular, since the LS-PLAN sequencing algorithm exploits LS, according to the Felder and Silverman's model [6], we used a KD where each LN is equipped with four weights, one for each FS dimension, representing the suitability of the LN for a given student. The domain parser, associated to the LS-PLAN sequencing algorithm, as shown in Section 2, acquires both LS weights and prerequisite relations from the LOM metadata of each LN and translates them in order to provide in output the correct input for the sequencing algorithm. Instead, the domain parser, associated to the TSA uses only one of the alternative learning materials linked to each concept.

While in Fig. 2 the empty GUI was shown, here in Fig. 4 we illustrate in the left part the GUI instance connected to the experiment, i.e., with the TSA and LS-PLAN algorithms, while in the right part of the figure is shown an example of how the GUI should appear in the case of the use of the actual IWT, Lecomps and KBS-Hyperbook algorithms.

We run the two algorithms on two different SM. The first, SM_1, is a learner that knows nothing about the domain. The second, SM_2, is a learner that knows the following concepts: *Neorealism origin*, *Rossellini*, *I bambini ci guardano*, *Rome in the Neorealism*, *De Sica*, *Topics*, *The War*, *Children*. The LOS produced by the two algorithms for the two SM are shown in Fig. 5. As we can see, the TSA did not manage the LS associated to the SM and to the LN: its LOS was built taking into account the *default* learning

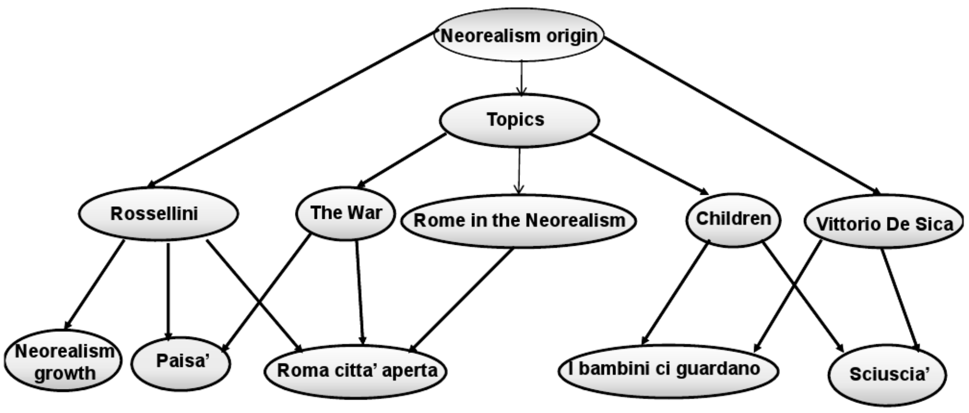
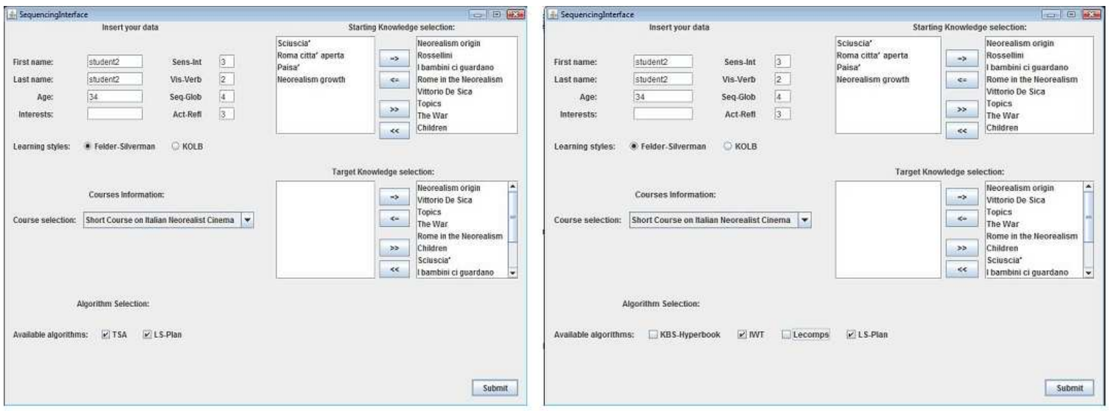


Figure 3: Italian Neorealist Cinema Domain.



First name: student2 | Sens-Int 3
Last name: student2 | Vis-Verb 2
Age: 34 | Seq-Glob 4
Interests: | Act-Refl 3

Learning styles: [✓ Felder-Silverman] KOLB

Course selection: Short Course on Italian Neorealist Cinema

Starting Knowledge selection: Sciuscia', Roma citta' aperta, Paisa', Neorealism growth | Neorealism origin, Rossellini, I bambini ci guardano, Rome in the Neorealism, Vittorio De Sica, Topics, The War, Children

Target Knowledge selection: | Neorealism origin, Vittorio De Sica, Topics, The War, Rome in the Neorealism, Children, Sciuscia', I bambini ci guardano

Available algorithms: [✓ TSA] [✓ LS-Plan]

First name: student2 | Sens-Int 3
Last name: student2 | Vis-Verb 2
Age: 34 | Seq-Glob 4
Interests: | Act-Refl 3

Learning styles: [✓ Felder-Silverman] KOLB

Course selection: Short Course on Italian Neorealist Cinema

Starting Knowledge selection: Sciuscia', Roma citta' aperta, Paisa', Neorealism growth | Neorealism origin, Rossellini, I bambini ci guardano, Rome in the Neorealism, Vittorio De Sica, Topics, The War, Children

Target Knowledge selection: | Neorealism origin, Vittorio De Sica, Topics, The War, Rome in the Neorealism, Children, Sciuscia', I bambini ci guardano

Available algorithms: KBS-Hyperbook [✓ IWT] Lecomps [✓ LS-Plan]

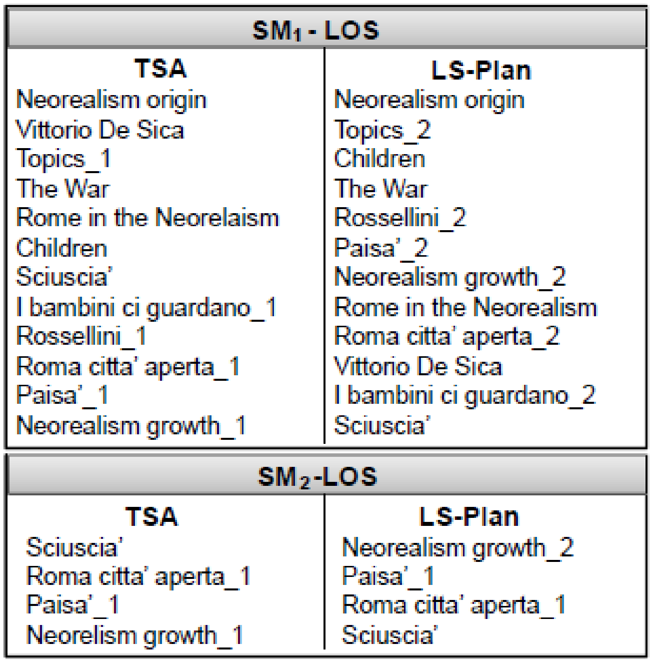
Figure 4: LS-Lab GUI.

material only, i.e., the learning material marked as suitable for every SM, shown in Fig. 5 with "_1". LS-PLAN selected, instead, different learning materials on the basis of the student's LS.

LS-LAB does not allow for a validation of the produced LOS: this is left to teachers and instructional designers. These experts can evaluate the suitability of the sequences for a given SM and their opinions can be stored, allowing a database of past evaluations.

4 Conclusions and Future Development

In this paper we presented the LS-LAB system, a framework, now at its early stage of development, that provides researchers with a suitable environment where to test, compare and validate their Curriculum Sequencing algorithms. We decided to develop such a



SM$_1$ - LOS	
TSA	**LS-Plan**
Neorealism origin	Neorealism origin
Vittorio De Sica	Topics_2
Topics_1	Children
The War	The War
Rome in the Neorelaism	Rossellini_2
Children	Paisa'_2
Sciuscia'	Neorealism growth_2
I bambini ci guardano_1	Rome in the Neorealism
Rossellini_1	Roma citta' aperta_2
Roma citta' aperta_1	Vittorio De Sica
Paisa'_1	I bambini ci guardano_2
Neorealism growth_1	Sciuscia'

SM$_2$-LOS	
TSA	**LS-Plan**
Sciuscia'	Neorealism growth_2
Roma citta' aperta_1	Paisa'_1
Paisa'_1	Roma citta' aperta_1
Neorelism growth_1	Sciuscia'

Figure 5: Produced LOS.

kind of system because of a lack in the literature of the Curriculum Sequencing research area of such an environment: while many sequencing algorithms have been presented during years, there is not a suitable environment where people can select a particular sequencing algorithm for a particular KD or compare different ones for research purposes. We proposed a first experiment based on two sequencing algorithms: a topological sorting-based algorithm vs. a linear temporal logic-based algorithm. The former has been used in several sequencing algorithms while the latter has been proposed in the LS-PLAN system, showing step by step the system together with the workflow to use it. As a future work we plan to implement other sequencing algorithms together with the possibility to communicate, through suitable communication protocols (e.g. SOAP) with remote algorithms, a better GUI and a XML binding in order to automatically input many semantic information such as the SM.

Acknowledgements

The authors wish to thank Marco Rosi, and Daniele Zelko who have contributed to various stages of the realization of this work.

References

[1] M. Baldoni, C. Baroglio, I. Brunkhorst, E. Marengo, and V. Patti. A service-oriented approach for curriculum planning and validation. In *Proceedings of International Workshop on Agents, Web-Services, and Ontologies - Integrated Methodologies, Durham, UK (6th–7th September 2007)*, pages 108–123, 2007.

[2] M. Baldoni, C. Baroglio, N. Henze, and V. Patti. Setting up a framework for comparing adaptive educational hypermedia: First steps and application on curriculum sequencing. In *Proc. of ABIS-Workshop 2002: Personalization for the mobile World, Workshop on Adaptivity and User Modeling in Interative Software Systems*, pages 43–50, Hannover, Germany, October 2002.

[3] P. De Bra, D. Smits, and N. Stash. Creating and delivering adaptive courses with AHA! In *EC-TEL*, pages 21–33, 2006.

[4] P. Brusilowsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11:87–110, 2001.

[5] P. Brusilowsky and J. Vassileva. Course sequencing techniques for large-scale web-based education. *International Journal of Continuing Engineering Education and Life-long Learning*, 13:75–94, 2003.

[6] Felder and Silverman. Learning and teaching styles in engineering education. *Engr. Education*, 1988.

[7] N. Henze and W. Nejdl. Adaptation in open corpus hypermedia. *International Journal of Artificial Intelligence in Education*, 12(4):325–350, 2001.

[8] C. Limongelli, F. Sciarrone, and G. Vaste. Ls-plan: An effective combination of dynamic courseware generation and learning styles in web-based education. In W. Nejdl, J. Kay, P. Pu, and E. Herder, editors, *Proc. of 5-th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2008)*, number 5149 in Lecture Notes in Computer Science, pages 133–142. Springer, 2008.

[9] Carla Limongelli, Filippo Sciarrone, and Giulia Vaste. An application of the ls-plan system to an educational hypermedia. *Int. J. of Web-Based Learning and Teaching Technologies*, 4(1):16–34, Jan-March 2009.

[10] M. Cialdea Mayer, C. Limongelli, A. Orlandini, and V. Poggioni. Linear temporal logic as an executable semantics for planning languages. *J. of Logic, Lang. and Inf.*, 1(16):63–89, Jan 2007.

[11] E. Sangineto, N. Capuano, M. Gaeta, and A. Micarelli. Adaptive course generation through learning styles representation. *Universal Access in the Information Society*, 7(1):1–23, 2008.

[12] A. Sterbini and M. Temperini. A logical framework for course configuration in elearning. In *Proc. of ITHET03*, July 2003.

[13] Stephan Weibelzahl and Christoph Ulrich Lauer. Framework for the evaluation of adaptive CBR-systems. In Ivo Vollrath, Sascha Schmitt, and Ulrich Reimer, editors, *Experience Management as Reuse of Knowledge. Proceedings of the 9th German Workshop on Case Based Reasoning, GWCBR2001*, pages 254–263, Baden-Baden, Germany, 2001.

[14] P. Wolper. The tableau method for temporal logic: an overview. *Journal of Logique et Analyse*, 28:119–152, 1985.

[15] Vladimir Zadorozhny, Michael Yudelson, and Peter Brusilovsky. A framework for performance evaluation of user modeling servers for web applications. *Web Intelli. and Agent Sys.*, 6(2):175–191, 2008.