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

Adaptive Learning with the LS-Plan System: a Field Evaluation

Carla Limongelli†, Filippo Sciarrone†, Marco Temperini‡, Giulia Vaste†

RT-DIA-141-2009 **Gennaio 2009**

† Università di Roma Tre,
Via della Vasca Navale, 79
00146 Roma, Italy
e-mail: limongel@dia.uniroma3.it, f.sciarrone@dia.uniroma3.it,
vaste@dia.uniroma3.it
‡ Department of Computer and Systems Science, University Sapienza, Roma, Italy
e-mail: marte@dis.uniroma1.it

ABSTRACT

LS-Plan is a framework for personalization and adaptation in e-learning. In such framework an Adaptation Engine plays a main role, managing the generation of personalized courses from suitable repositories of learning nodes and allowing the maintenance of such courses, for continuous adaptation of the learning material proposed to the learner. Adaptation is meant, here, with respect to the knowledge possessed by the learner and her learning styles, both evaluated prior of the course and maintained during course taking. Knowledge and Learning styles, are the components of the student model managed by the framework. Both the static, pre-course, and dynamic, in-course, generation of personalized learning paths are managed through an adaptation algorithm and performed by a planner, based on Linear Temporal Logic. A first Learning Objects Sequence is produced, basing on the initial learner's Cognitive State and Learning Styles, as assessed through pre-navigation tests. During the students navigation, and on the basis of learning assessments, the adaptation algorithm can output a new Learning Objects Sequence, to respond to changes in the student model. We report here also on an extensive experimental evaluation, performed by integrating LS-Plan in an educational hypermedia, the Lecomps web application, and using it to produce and deliver several personalized courses in an educational environment dedicated to *Italian Neorealist Cinema*. The evaluation is performed following mainly two standard procedures, the *As a Whole* and the *Layered* approaches. The results appear to be encouraging on both the system on the whole and the adaptive components.

Contents

1 Introduction

Modern research in hypermedia systems focuses most attention on adaptivity. As pointed out in [6], adaptive hypermedia systems are developed in opposition to the traditional 'one-size-fits-all' approach, allowing both user modelling and adaptation to the current user's needs. Educational hypermedia are one of the most important application field of adaptive technologies.

In this paper we present the LS-Plan system [28, 29], a system capable to provide educational hypermedia with adaptation and personalization, together with an extensive empirical evaluation.

Personalization and adaptation in educational systems are often associated with the *Course Sequencing*, that is producing an individualized sequence of didactic materials or activities for each student, dynamically selecting the most appropriate ones at any moment [7]. In this context, one widely used approach is *Dynamic Courseware Generation* [7], where the personalized course is generated so to lead the learner from her initial state of knowledge, till to cover a stated set of learning goals, and the course content is adapted to the learner's progress. We can classify course sequencing techniques into two categories:

- sequencing that plans the entire learning path at the beginning, then modifies it, when the study does not succeed as it should (e.g. DCG [7], the work of Baldoni et al. [3, 4] and the IWT system [27]).
- sequencing obtained in an implicit way, step-by-step, through adaptive navigation support techniques, such as adaptive link annotation and direct guidance [6] (like the AHA! System [17] and the ELM-ART system [38]).

In both approaches, personalized sequencing is generally mainly based on the student's knowledge, modelled as an *Overlay Model* of the domain of interest [9]. Learning, however, involves also a lot of psychological features, such as motivation, Cognitive State (CS) and learning styles (LS). These features and the ways of how to produce systems that use them effectively, are presently investigated in several works. In [9], opposite opinions about the effectiveness of adaptation based on LS are illustrated. However a lot of studies have been carried out in considering LS for modelling students, based on the idea that teaching strategies, based also on the student LS, might increase the effectiveness of learning and motivate learners. Felder and Silverman's Learning Styles Model (FS) [19] has been often taken into consideration in the literature ([27, 22, 2, 11, 1]) and Zywno showed its effectiveness in engineering educational context [41].

In this paper we evaluate the effectiveness of LS-Plan, a system that provides sequencing both at *static* level, by planning the whole learning path ahead of the course, and at *dynamic* level, by rebuilding the path, possibly step by step, during the course taking. The system also guides the learner during the course, through provision of alternative didactic material. The system models the student with LS, and, from the cognitive point of view, through an overlay model. LS are modelled according to the Felder and Silverman's Learning Styles Model. The Cognitive State (CS) is represented by a set of Knowledge

Items (KI), i.e., atomic elements of knowledge concerning the learning domain, according to the Knowledge Space Theory [18]. The KI are associated with one of three levels of Bloom's Taxonomy [5]. LS-Plan is based on the synergy between classical planning techniques and LS refinement procedures, considering LS as tendencies that may change during the educational experiences [20]. The main component of LS-Plan is an adaptation algorithm, capable to modify the student's model on the basis of self-assessments and navigational behavior, and to consequently guide her step by step, especially in the recovery activity. At the same time the system lets the student free to navigate in the learning hyperspace.

LS-Plan is composed by three main modules: the Adaptation Engine, the Planner and the Teacher Assistant. The last one is responsible for teacher's functionalities; the Planner takes in input the student model, the domain model and teacher's didactic strategies producing as output a Learning Objects Sequence (LOS), that is the personalized course. The Adaptation Engine manages the User Model and the Adaptation Decision Making [8].

In order to evaluate the LS-Plan system, we embedded it into a web application, the Lecomps system [37], able to manage the educational environment of *Italian Neorealist Cinema* to support our experiments. We managed to involve a set of 30 individuals in the experiment, by means of a standard sample selection procedure. After we performed two main evaluations: the *As a Whole* and the *Layered* evaluations, following the guidelines proposed in the literature of experimental evaluations of adaptive systems [8, 12, 21]. In the first evaluation, participants were partitioned into two groups, one using the basic hypermedia, i.e., without the adaptive features provided by LS-Plan , the other using the Lecomps-LS-Plan integrated system. The Layered evaluation aimed to check the adaptive components separately. In particular we studied separately the aspect of user modelling and of adaptation decision making. As of the user modelling aspect, we allowed the learner to constantly monitor her own LS evaluation and known concepts. In this paper, we also evaluated the quality of navigation patterns followed by the learners: *How often the learners followed the recommendations provided by the system?*, *How often they got out of the suggested path?*, and *What were the comparative results of such behavior in terms of assessment and study time?*

The rest of the paper is organized as follows: Section 2 gives a description of the related work. Section 3 illustrates the architecture of LS-Plan together with its main components. Section 4 provides a short description of the Lecomps system embedding the LS-Plan during the experiments. Section 5 reports on the experimentation ratio and findings. In 6 our conclusions are drawn.

2 Related work

In this section we report on the work related to the two approaches to sequencing we mentioned in Sec.1. We will firstly discuss the "sequencing/re-sequencing" approach, which is based on an initial production of a complete sequencing, followed by successive re-sequencing actions when and if it is the case; we will then report on the "implicit/step-by-step" approach, based on adaptive navigation support techniques.

We then report also on trends in the management of learning styles for adaptivity.

2.1 Sequencing/Re-sequencing

LS-Plan produces the learning path at the beginning of the course through the PDK planner [13]. The approach to modeling course sequencing as a planning problem is very similar to the one adopted in [3, 4], in which learning resources (*learning objects* in [3] or *courses* in [4]) are seen as actions, with preconditions and effects, i.e. with prerequisites and acquired competencies, specified in the "Classification" tag of the IEEE LOM standard [30]. The definition of these metadata is based on ontologies of interest to guarantee shared meanings, interoperability and reusability, allowing for a Semantic Web perspective. However in these approaches, "tagging" is a bottleneck: teachers may find it hard to adhere to predefined ontologies. Moreover, in [3, 4] personalization is not performed at the level of learning materials, so the teacher cannot express how to choose the most appropriate learning object among those that explain the same concept. Finally we could see that LS-Plan has some commonalities with DCG [7] that creates a plan of the course contents, follows the student during the fruition of the course, and makes a re-planning if the student fails to demonstrate the acquisition of a concept. Sequencing in DCG is sophisticated, and considers some personal characteristics though, to our knowledge, it does not let re-sequencing actions depend on the occurring learning styles modifications.

2.2 Implicit/Step-by-step

Adaptive navigation support techniques are widely used in AHA! [17] and in ELM-ART [38]. AHA! is a very flexible system, where adaptation can be performed both through navigation support and in contents, including *fragments adaptation* [6]. It is based on rules, managing both user modeling and adaptation strategies. The management of such rules, and in particular their termination and confluence, might be a drawback in AHA!, in fact it guarantees termination through enforcements, while the confluence problem is left open (see [40] for a complete dissertation about these problems). Another drawback is then related to producing a specification of such rules, suitable for their use in the system. So, significant efforts are presently devoted to the development of advanced authoring tools: for example MOT [16] allows authoring based on LAOS [14] and LAG [15] models, making it possible to use an adaptation language to program adaptive behaviours, which will be compiled in suitable rules. However, authors are required either to possess programming skills or to rely upon pre-defined strategies. Moreover AHA! does not exploit assessment for adaptivity so far. ELM-ART has, instead, a knowledge domain management similar to the LS-Plan one: the teacher can define the prerequisites and tests related to concepts. The student model, from a cognitive state point of view, is more granular, taking into account inferences and concepts marked as known by the student. It does not consider, however, learning styles.

2.3 Learning Styles

Learning styles, and in particular the Felder and Silverman's model, are used in a lot of systems, such as the following:

- the add-on for the Moodle Learning Management System proposed in [22], where a course personalization, based on LS, is presented;
- the system proposed in [2], in which an interesting adaptive interface is included;
- the Tangow system [1] that uses two dimensions of the Felder and Silverman's Model; it initializes the student model in an explicit way, through the Felder and Soloman's Index of Learning Styles (ILS) Questionnaire; updates such model in an implicit way, through observations of the student's browsing behavior, and uses the model information also to encourage collaborative learning through group formation;
- the CS383 system [11] and the IWT system [27] that propose an adaptive presentation based on learning material typologies.

LS-Plan learning styles management is finer grained, because the system allows teachers to assign different weights to the actual learning material, not only to its typology, according to the four Felder-Silverman's LS dimensions. In this way the system provides the teacher with the possibility to implement different didactic strategies for different learners. Moreover LS-Plan, as well as Tangow, takes into account the information gathered from the student's behavior, but, differently, it considers the information derived from both navigation and self-assessments, in order to evaluate the effectiveness of the current teaching strategy, and modifying it is necessary.

3 The Adaptive System

Fig. 1 shows the overall system; LS-Plan provides the educational hypermedia with adaptivity; the main components are highlighted with grey blocks and described in the following.

The *Teacher Assistant* is responsible for the teacher's functionalities. It allows the teacher to arrange a pool of learning objects, i.e., learning nodes, that is to define all the metadata necessary to tag such materials. This information is stored in a database, belonging to LS-Plan, while the actual repository of learning material is stored in the educational hypermedia. The Teacher Assistant allows also the teacher to define tests related to learning nodes, and to create the initial *Cognitive State Questionnaire* for evaluating the starting knowledge of the student, that is the knowledge already possessed by the student with respect to the topic to be learned. The student fills in both the *Cognitive State Questionnaire* and the *Index of Learning Styles (ILS) Questionnaire*, i.e.

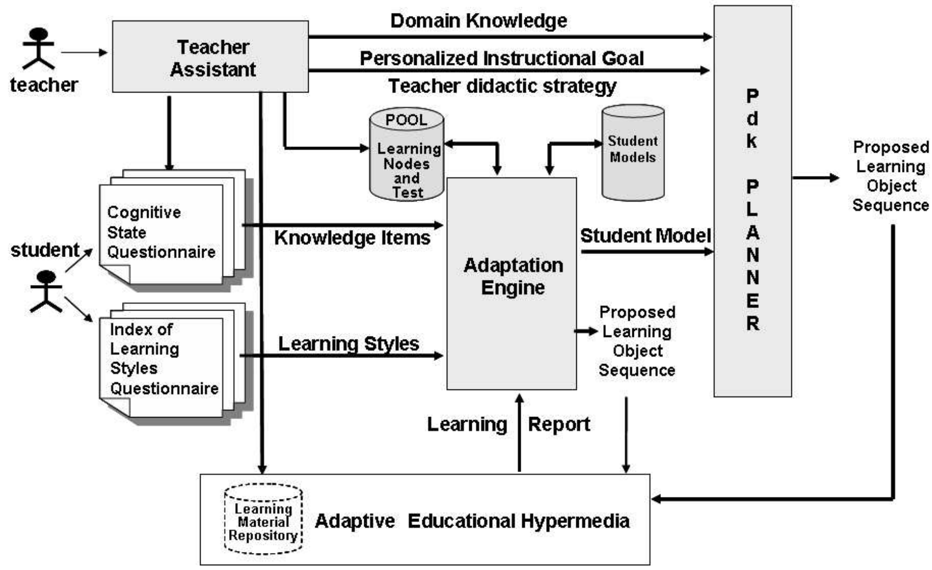

Figure 1: The Functional Schema of the Adaptive System. Grey blocks form LS-Plan.

a test, developed by Felder and Soloman [1], that extracts the student's learning preferences according to the four dimensions of the FS Model: active-reflective, sensing-intuitive, visual-verbal, sequential-global [19]. This information is managed by the Adaptation Engine, in order to initialize the student model, which is then stored in the *Student Models Database*. Through the Teacher Assistant, the teacher specifies also her didactic strategies and defines for each student her own instructional goal. This information together with both the results of the two initial questionnaires and the descriptions of the learning nodes, i.e. the Domain Knowledge, are coded in .pddl files and sent to the *Pdk Planner*.

The *Pdk Planner* produces in output to the hypermedia a personalized LOS for the given student. The student is not forced to follow the LOS generated by the planner.

The *Adaptation Engine* follows the student's progresses during the fruition of the course, taking into account results from intermediate questionnaires and the time spent in the study of each learning node. This information is used both for updating the student model and for the adaptation decision making, as it will be discussed in Section 3.1.2.

Before going into details about the components of the system, the algorithms used for managing the student model updating, and the adaptation decision making, we introduce some definitions about the elements we are going to work with.

Definition 3.1 (Knowledge Item). *A Knowledge Item KI is an atomic element of knowledge about a given topic. KI is a set:*

$$KI = \{KI_K, KI_A, KI_E\}$$

where KI_ℓ, with $\ell \in \{K, A, E\}$, represents a cognitive level taken from Bloom's Taxonomy: Knowledge, Application and Evaluation.

[1]Available at http://www.engr.ncsu.edu/learningstyles/ilsweb.html

Definition 3.2 (Learning Style). *A Learning Style LS is a 4-tuple:*

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

where each D_i is a FS Learning Style Dimension, i.e., D_1: active-reflective, D_2: sensing-intuitive, D_3: visual-verbal, D_4: sequential-global.

We used the range $[-11, \ldots, +11]$ according to the Felder-Soloman ILS scale.

Definition 3.3 (Learning Node). *A Learning Node LN is a 5-tuple:*

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

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

AK *Acquired Knowledge. It is a KI_ℓ, with an associated success threshold σ_{KI_ℓ} defined in Definition 3.4, that represents the knowledge that the student acquires at a given level as specified in Definition 3.1, after having passed the assessment test related to the KI_ℓ of the node. If such a test is not present in the node the AK is considered acquired anyway.*

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

LS *is given in Definition 3.2.*

T *is a pair of reals $T = (t_{min}, t_{max})$ which represents the estimated time interval for studying the material of the node, as prefixed by the teacher. A fruition time, t_f, less than t_{min} is not a realistic time to learn that material; for a fruition time, t_f, greater than t_{max} we have the so-called "*coffee break*" effect, i.e., the student is supposed to have done something else.*

Definition 3.4 (Threshold σ_{KI_ℓ}). *A threshold value σ_{KI_ℓ} is a real number associated to KI_ℓ, defined as:*

$$\sigma_{KI_\ell} = \frac{S_T}{S_{max}}, \qquad 0 < \sigma_{KI_\ell} \leq 1$$

being S_T the lowest score of an assessment test, as fixed by the teacher, in order to consider the KI_ℓ acquired; S_{max} is the highest possible score for that test [2]*.*

Definition 3.5 (Pool). *A pool is the particular set of LN, selected or created by the teacher in order to arrange a course about a particular topic.*

Definition 3.6 (Domain Knowledge). *The Domain Knowledge DK is the set of all the KI present in a pool.*

[2]Here and in the following we suppose that the teacher gives to S_T and S_{max} positive values.

Definition 3.7 (Cognitive State). *The Cognitive State CS is the set of all the KI_ℓ possessed by the student with respect to the given topic: $CS \subseteq DK$.*

Definition 3.8 (Student Model). *The Student Model SM is a pair:*

$$SM = (CS, LS)$$

where, CS is given in Definition 3.7 and LS is given in Definition 3.2.

Definition 3.9 (Test). *A Test is a set of k items, i.e., questions, with $k \in \mathcal{N}$. To each item is associated a weight $Q_j \in \mathcal{R}$. Each item has m answers, with $m \in \mathcal{N} - \{1, \infty\}$ and to each answer is associated a weight $w_i \in \mathcal{R}$.*

Let S_{KI_ℓ} be the score associated to a test; it assesses the student knowledge of the single KI_ℓ:

$$S_{KI_\ell} = \sum_{j=1}^{k} (Q_j \cdot \sum_{i=1}^{m} w_i)$$

where $w_i = 0$ for the answers the student does not select.

Definition 3.10 (Acquisition of a KI_ℓ). *A KI_ℓ is supposed to be acquired by the student if:*

$$S_{KI_\ell} = \sum_{j=1}^{k} (Q_j \cdot \sum_{i=1}^{m} w_i) \geq \sigma_{KI_\ell}$$

where σ_{KI_ℓ} is given in Definition 3.4 and k, m, Q_j, w_i and S_{KI_ℓ} are given in Definition 3.9.

3.1 The Adaptation Engine

In this section we show the mechanisms of the SM management, i.e., the initialization and the updating processes, and the related adaptation strategies.

3.1.1 Student Model Initialization

At the first access to the system the student fills in the *Cognitive State Questionnaire* composed by some tests (see Def.3.9), related to the KI of the Domain. The acquisition of a KI_ℓ is described in Def.3.10. All the acquired KI_ℓ initialize the CS, which can also be an empty set, if the student does not know anything about the domain. The student also fills in the *ILS Questionnaire* whose results are used to initialize her own LS.

3.1.2 Student Model Updating and Adaptation Methodology

A revised version of the student model updating and adaptive decision making algorithms presented in [28] have been proposed in [29]. Here we summarize the steps of the algorithms that are the core of the system we are going to experiment. At each step of the learning process, i.e., after the student studies the contents of a Learning Node, the algorithm carries out two main actions:

UPDATESM (LearningNode LN , StudentModel SM_{old})

$t_f \leftarrow$ TIMESPENTONTHENODE (LN)

$S_{KI_\ell} \leftarrow$ COMPUTESCOREPOSTTEST (KI_ℓ)

/* Learning Styles Updating */

if (not post-test) **then**

  $LS_{new} \leftarrow LS_{old}$

else

  if (KI is acquired) **then**

    for all D_i

      $D_{i_{new}} = D_{i_{old}} + \eta_1(t_f, S_{KI_\ell}) sign(\Delta D_i)$

  else

    for all D_i

      $D_{i_{new}} = D_{i_{old}} - \eta_2(t_f, S_{KI_\ell}) sign(\Delta D_i)$

  $LS_{new} \leftarrow \langle D_{1_{new}}, D_{2_{new}}, D_{3_{new}}, D_{4_{new}} \rangle$

/* Cognitive State Updating */

if ((not post-test) or (KI acquired)) **then**

  $CS_{new} \leftarrow CS_{old} \cup AK$

else

  $CS_{new} \leftarrow CS_{old}$

$SM_{new} \leftarrow (CS_{new}, LS_{new})$

return SM_{new}

Figure 2: The function UpdateSM.

- the updating of the student model;
- the computation of the Next Node to be proposed to the student, together with the new Learning Object Sequence.

Basically the idea is to work like the teacher would do: re-explaining the failing concept (proposing the same learning material as before), then trying to propose different learning material for the some concept, and finally, on further fail, assuming that some of the prerequisite, previously given for granted, are the source of the problem and will be suggested for re-checking.

Fig.2 and Fig.4 present the algorithms related to these two actions, respectively.

When the student studies a LN, the function UPDATESM is activated taking in input the LN and the current SM. The function TIMESPENTONTHENODE computes and

act.-refl.	sens.-int.	vis.-verb.	seq.-glb.
-1	-5	-3	-3
-0,15339	-4,15339	-2,15339	-2,15339
0,716102	-3,2839	-1,2839	-1,2839
1,358051	-2,64195	-1,92585	-0,64195
0,736441	-3,26356	-1,30424	-1,26356
1,402825	-2,59718	-0,63785	-0,59718
0,902825	-3,09718	-0,13785	-1,09718
1,733757	-2,26624	-0,96879	-0,26624
2,428672	-1,57133	-0,27387	0,428672
1,820763	-0,96342	0,33404	1,036582
2,491384	-1,63404	-0,33658	0,36596

Table 1: A student LS evolution

returns the time t_f, that is the time spent by the student on the node. The function ComputeScorePostTest computes and returns the score taken by the student in the post-test related to the KI_ℓ of the node, that is the AK related to the LN. In case the post-test is not provided we can only assume a 0 score and consider the KI_ℓ related to that node as acquired, though without updating the student's LS. On the other hand, if the post-test is available, we update the student's LS according to the LS associated to the node, to the fruition time t_f, and to the score obtained with the post-test. In particular, if the KI_ℓ is acquired, we consider that learning material adequate to the case, so we can update the student's LS towards the LS of the node, by an extent depending directly on score and inversely on time t_f. On the contrary if the KI_ℓ is not acquired we apply the opposite behavior. Let us note that such modifications are to be considered as "adjustments" for the present LS estimate, so they are actually quantified as values in $[0, ..., 1]$. In order to "quantify" this updating process we introduce two functions, α and β, that normalize η_1 and η_2 in $[0, ..., 1]$, as shown in [29]. For example, Tab.1 and Fig.3 show a student LS evolution, taken from the log files of our experimentation.

The function NextNode proposes the next node to be learned on the basis of the new SM as described in Fig. 4. If a D_i, as given in Def.3.2 changes sign, we consider to be in presence of a significant variation in the student LS, which makes it necessary to re-plan the LOS: the algorithm suggests the first LN of the new LOS computed by the planner. If the student does not pass the test, the time t_f is examined: the boolean function "time-out" checks whether t_f is out of range and if it is the first time that the LN has been studied. In positive case, the system proposes once again the same node to the student. After the second unsuccessful trial, the system applies the function CheckClosestNode, that looks for the closest node, computed by selecting an alternative node, LN' with the same RK and the same AK of the current LN, that is the node with the smallest distance to the student's LS, computed on the basis of

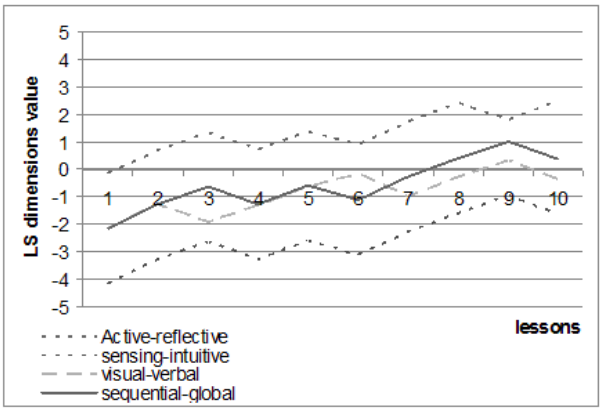

Figure 3: An example of student LS evolution.

the Euclidean distance metric. If such a node does not exist, by means of the function OrderedPredecesorsList, the algorithm computes the list L of the LN predecessors, i.e., the nodes connected to the LN by an incoming link, in order to verify the acquisition of prerequisites, RK, related to the LN. The nodes are accommodated in the list L, according to the following precedence classes:

1. the predecessor nodes that have not been visited by the student. In fact it is possible that the student got the AK related to that node, by giving a correct answer to the initial test, but she lacks that concept indeed;

2. the nodes that do not provide tests are proposed on the basis of the difficulty levels: K, A, E;

3. the nodes that provide test whose LS are closest to the student's LS.

The AK of the prerequisite nodes, if present, are removed from the CS, because we are in presence of a sort of "loss" of knowledge. Then the algorithm puts L on the top of the LOS and suggests its first LN. If both the attempts to explain the concepts with different learning material and the prerequisite checks fail, the algorithm re-plans a new LOS and proposes its first. If no plan is found, we assume that all the possible LOS have been already taken into account: this is the case when the teacher has to regain control.

3.2 The Teacher Assistant

The *Teacher Assistant* is responsible for the management of the functionalities provided for the teacher, i.e., for the management of the pool. The teacher also selects the items and the threshold for the *Cognitive State Questionnaire* and manages the students' registration to the course. In particular she decides the student's instructional goal and specifies her didactic strategies, such as the desired level of the course, or the particular way she prefers to explain a given concept.

NextNode (LearningNode LN , StudentModel SM_{new})

if ($\exists D_i$ that changed sign) **then**

  Replan(SM)

  return (LN^{first}, SM)

else

  if (time-out(t_f)) **then**

    return (LN, SM)

  else

    $LN' \leftarrow$ CheckClosestNode (LN , SM)

    if ($LN' \neq NIL$) **then**

      return (LN', SM)

    else

      $L \leftarrow$ OrderedPredecessorsList (LN)

      if ($L \neq NIL$) **then**

        $\forall LN_i \in L$, **if** ($AK \in CS$) **then**

          $CS \leftarrow CS - AK$

        Add_L_ToPlan (L)

        return (LN^{first}, SM)

      else

        Replan(SM)

        return (LN^{first}, SM)

Figure 4: The function NextNode.

3.3 The Pdk Planner

Here we describe how automated planners, in particular the logic based ones, can support either one of the processes of course configuration and domain validation. In the context of course configuration, planning problems are described by actions (LN), specifying action preconditions (RK) and action effects (AK), as well as the initial state (initial SM) and the goal. Besides all these basic elements a teacher would be allowed to express her didactic strategies, e.g., preferences related to a concept explanation. To this aim we use the planning language PDDL-K and the Pdk planner (Planning with Domain Knowledge) [13][3]. It conforms to the planning as satisfiability paradigm: the logic used to encode planning problems is propositional Linear Time Logic (LTL). PDDL-K [13], conforming to standard PDDL, guides the teacher, through the Teacher Assistant, in the specification of heuristic knowledge, providing a set of control schemata, that is a simple way of expressing control knowledge. The language is given an executable semantics by means of its translation into LTL. In the following we sketch the Pdk functionalities that are exploited by LS-Plan; for a more detailed description see [29]

- **Course sequencing:** the initial conditions are given by the CS of the initial student model; the goal of the problem corresponds to the course target knowledge. In this way course sequencing is the synthesis of the actions that the planner produce to reach the goal.

- **Domain validation:** this is the check about pool consistency, conforming to the style of [17].The loop check is an easy control for the planner: the SK is the empty knowledge and the TK is the set of all KI. The control discovers actions that can never be executed.

- **Redundant formula detection:** this phase can help the teacher in arranging the pool of the LN.

- **Heuristic control knowledge specification:** the PDDL-K specification language provides a set of control schemata that allows the teacher to set some didactic strategies such as: the desired level of difficulty (see Def.3.1), the particular way the teacher prefers to explain a given concept, or the the constraint about the execution of some actions the teachers believes mandatory for all the students, even if they demonstrate to know the concepts related to them.

We have to notice that, although automated planning is computationally a hard task, the practical execution time depends on many variables, such as the number of pre and post conditions of the actions and the number of goals [10]. Moreover, the definition of correct control knowledge is also a difficult task and can generate inconsistent problems. However, the high-level control formulas provided by PDDL-K provide a set of predefined schemata and allows one to easy and naturally specify heuristic knowledge, without requiring specific programming skills. An appropriate heuristic knowledge can prune the search space and

[3]Available at http://pdk.dia.uniroma3.it/

can improve the performances of the planner, both in terms of execution times and plan quality. From a practical point of view, our experiments presented in [13] shown that pools with up to 100 nodes can be managed by the planner in less than 5 secs.

4 LecompS: a web application embedding LS-Plan

As it was seen above, LS-Plan provides learners and teachers with a framework organizing the generation of personalized courses: LecompS is the web application that enables the delivery of such courses acting as the educational hypermedia.

Such a web application is the evolution of a system implementing a logical framework for automated course production [36]; it's widely evolved to support also the functional characteristics related to LS-Plan. As a full description of the system is not in the focus of this paper, we give some very brief discussion of it in the following three subsections, concluding then by just introducing the experimental evaluation described in Sec.5.

4.1 Educational environment

This is the basic organization of the learning material related to a given subject matter. Each subject matter is defined by a set of information, describing ratio and content of the course, and two more crucial components: the repository (didactic pool) of learning nodes to be used for the construction of courses, and the definition of a target knowledge that will be common to all the courses built under such subject matter. When an individual course is to be constructed, the system will access the initial state of learner's cognitive state, and the environment's pool and target knowledge.

In the repository of learning nodes, associated to a given educational environment, the learning nodes can be created and edited, through a web based editor (FCK editor). The pool can be accessed also through a graphical interface, allowing to see the propaedeuticy relationships among the learning nodes, and visualize and/or edit them.

4.2 Enrolment in an educational environment and course delivery

These are learner's-oriented functionalities. A prospective learner can see the information related to the educational environment, enrol in it and submit the questionnaires to input the system with her initial cognitive state, as related to the subject matter, and learning styles evaluation.

When the personalized course is available (see next subsection) the learner can access an take the learning material. Two examples of access page to a course are shown later on, in Fig.6 and Fig.7.

4.3 Course construction

When for a learner the initial cognitive state and learning styles are available, the process of automated configuration of the course via LS-Plan can be activated. Although also such activation can be made automatic, the present implementation of the system leaves it "on the finger of the teacher" (Fig.5), allowing to possibly modify, before of activation, some parameters (such as the overall effort of a lesson and the starting cognitive state).

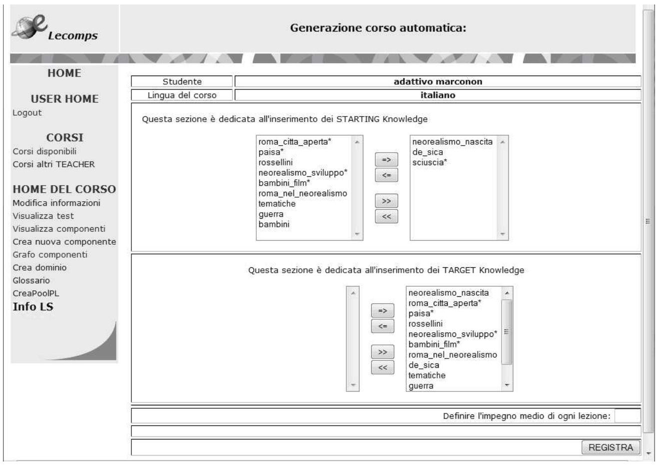

Figure 5: Teacher's interface for activation of course construction: the button activates the automated configuration of a course; both the initial cognitive state (starting knowledge) and the target knowledge are modifiable here (not expected to be done usually)

The experimental evaluation described in the next section is based on the use of two different versions of LecompS, one allowing the full application of the LS-Plan framework and another, providing a non-adaptive management of courses.

In Fig.6 the interface used by a learner to take adaptive course is shown. The upper part of the figure gives the sequence of learning nodes stated for the learner in accord with her initial cognitive state and learning style evaluation. This is the actual personalized course, listing all the prescribed learning nodes. On the other hand, the whole set of learning nodes available in the educational environment's pool are available to the the learner in the lower part of the page, to allow access to the learning material also in a non-prescribed manner.

The course is taken by selecting one learning node at time (the small books in figure are links to learning nodes). After each learning node, the learner can take an assessment test; on the basis of the answers to the test the student model can be updated and the course can be possibly adapted. Feedbacks to such update are twofold: as a consequence of modifications in the cognitive state, the learner can see changes in the sequence of learning nodes for her course (only the learning nodes still to be taken towards course termination are listed in the upper part of the page); as of learning styles and cognitive state modifications, they can be appreciated by accessing a related page, where the learner can see a discursive description of the present state and grade the agreement towards such

Figure 6: Learner's page for a course during the experimentation described in Sec.5 (adaptive management).

an evaluation.

In Fig.7 the learner's interface for non-adaptive courses is shown: this is basically the list of the learning nodes to be taken, with no further treatment by the system.

5 Evaluation

In this section we show an empirical evaluation of the LS-Plan system, obtained by experimenting its embedding into the LecompS Hypermedia system. As a matter of facts, our main research questions, here, concern the reliability and the value of our adaptative framework, once it is applied to the educational hypermedia system. In our experimentation we followed the guidelines for the empirical evaluation of adaptive systems embedded in interactive hypermedia systems, as outlined by Chin [12], Brusilovsky [8], Gena [21] and Masthorff [31].

This section is structured as it follows: in Sec.5.1, we show the experimental environment setup, where all the parameters needed to start the experiments are drawn. In Sec.5.2 we propose the *As a Whole* experimentation of the system, i.e., a controlled *within subjects* experiment performed in order to test learners' learning in a *with Vs. without*

Figure 7: Learner's page for a course during the experimentation described in Sec.5 (non-adaptive management).

modality. In Sec.5.3, a *Layered Evaluation* ([8]) of the adaptive components (both *SM* and *Adaptation Decision Making* process) is performed. Finally, in order to take into consideration the degree of learner's agreement and appreciation in their interaction with the system, Sec.5.4 draws a *Happy Sheet* Analysis, and illustrates a statistical description of the navigation parameters. The results of the three evaluations are discussed separately in each section.

5.1 Experimental Setup

In this section we show the experimental environment we built to run the experiments, beginning from the selected knowledge domain and available at `http://paganini.dia.uniroma3.it:8080` web site together with all the statistical data[4].

5.1.1 The Knowledge Domain

We used the Lecomps hypermedia system in order to teach topics on *Italian Neorealist Cinema*. A film critic has also been involved in the project, as the domain expert and educational expert. The Neorealist period was a very particular time span for Italian cinema, more or less situated between 1945 and 1952, altought it's difficult to pinpoint precise temporal boundaries in such fields. Those were the post-war years of reconstruction, both material and moral reconstruction and above all of Italy's assertion through a crude testimony of social conditions, the inevitable starting point for any transition.

[4] This server is a departmental protected server and a password is needed. Contact authors for login.

We chose this domain in order to be able to run in the future large-scale experiments in humanistic fields as well. We built a knowledge domain composed of 18 LN, each of them having an associated test and of 12 KI, selected by the domain expert.

5.1.2 Questionnaires

In order to evaluate the student knowledge and satisfaction, the following questionnaires were built and proposed:

- Pre-test questionnaire. A questionnaire formed by 50 questions to check the starting knowledge of the sample.
- Post-test questionnaire. A questionnaire formed by 50 questions which aimed to measure the post-navigation knowledge of students.
- LN-questionnaire. A questionnaire for every LN was prepared in order to evaluate the acquired knowledge after having visited it.
- Happy Sheet. This is a questionnaire formed by 13 questions, devised in order to evaluate students' degree of satisfaction in the use of the system, submitted as the last step of the experimentation.

5.1.3 The Sample

The sample was randomly selected among students, both from University, high schools and teachers, who were interested to learn something about the domain proposed by the LECOMPS system. The process of sample gathering has been divided in several steps. In the first step we selected about 45 individuals. In the second step, in order to have a homogeneous starting group (that is a group enjoying the same average a-priori knowledge about the learning domain) we gave the whole group a pre-test questionnaire containing questions about the most important issues addressed by the learning domain. In the third step we formed a homogeneous group with the lowest average, i.e., the lowest starting knowledge on the domain and the lowest dispersion around it. We obtained one group of individuals with average $\overline{x} = 6.81$ and standard deviation $S_{\overline{x}} = 4.36$. We considered these two values as a good compromise to have both a low starting knowledge and a low dispersion, in order minimize the well known statistical problem due to the *between subjects dispersion* [12, 31]. In this way, we reduced our sample to 30 users, equally distributed between males and females and with age in the range $[20, 50]$. In particular, on the average, our sample started with a domain knowledge of 16.23% , i.e., every individual obtained, on average, the 16.23% of the maximum possible score [5].

[5] Actually, because of the standard deviation $S_{\Delta_S} \neq 0$, in the hypothesis of normal distribution for small size samples, by means of the t-student stochastic variable users could obtain up to the 24.61% of knowledge.

5.2 The *As a Whole* Evaluation

In this Section we show the double blind controlled experiment in the *with* Vs. *without* adaptive component, i.e., the LS-PLAN engine, in order to investigate our research question.

5.2.1 Research Question

The Research Question (RQ) is:

Does students navigating with the adaptive modality learn more than students navigating without the adaptive modality?

5.2.2 Statistical Model

In order to answer to our research questions, we exploited the hypothesis testing technique. To this aim, we adopted the following working assumptions:

- Independent Variable. We defined the independent variable Δ_S as the difference between the score obtained by each student in the post-navigation test S_{post} and the one obtained in the pre-navigation test, S_{pre}:

$$\Delta_S = S_{post} - S_{pre} \tag{1}$$

  This independent variable allows us to measure the real improvement shown by the student after her learning.

- Use of the distribution-free statistics. We did not assume that the statistical distributions which our independent variable Δ_S belongs to is the normal distribution (e.g. [35, 25]).

- Use of the Test of Wilcoxon-Mann-Withney (WMW) for two independent samples. This test, coming from the psychological research area, is well suited to testing in experiments where humans play a crucial role [39]. To this aim we divided the sample into two groups, the experimental group X, formed by 15 individuals, who navigated with the adaptivity modality and the control group Y, formed by 15 individuals, who navigated without the adaptive modality. We indicate with Δ_X and Δ_Y the values of the variable Δ_S respectively for the group X and Y.

5.2.3 Data gathering

Students of both groups were required to navigate into the system. On the basis of our work hypotheses, we gathered all the pre-Navigation Scores S_{pre} and all the post-Navigation Scores S_{post}, in order to compute the variable Δ_S. In Tab.2 the main statistical parameters are shown.

Modality	**St. Dev.** S_{Δ_S}	$\overline{\Delta_S}$
Adaptive	4.06	10.30
Non-Adaptive	4.29	7.61

Table 2: The Statistical data.

n	m	$\overline{W_x}$	S_{W_x}	Z	$p - value$
15	15	232.5	581.25	1.82	0.03

Table 3: The Wilcoxon Test Parameters.

As we can see, on the average, students who navigated with the adaptive environment, showed a higher improvement than students who navigated in the non-adaptive environment.

5.2.4 Hypothesis Testing

Here we show the WMW testing procedure applied to our statistical data. Firstly we define the Null Hypothesis H_0: the two learning modalities show a not significative statistical difference between them and the variables Δ_X and Δ_Y belong to the same statistical distribution, i.e., no difference between the two learning modalities. Secondly, we define the Alternative Hypothesis H_1: the two learning modalities show a significative statistical difference between them and the variables Δ_X and Δ_{LY} belong to different statistical distributions, i.e., there is difference between the two learning modalities. Thirdly, we fixed our significance level $\alpha = 0.05$. Finally, following the standard WMW procedure, we obtained: $p - value = 0.03$, as shown in Tab.3.

As we can see, we obtained $p - value < \alpha$. In this way, we can reject the Null Hypothesis H_0 and accept, the alternative Hypothesis H_1. The differences are not due to chance.

5.2.5 Discussion

Here we point out the statistical results of the *On the Whole* evaluation. This evaluation showed that the two independent variables Δ_X Δ_Y belong to two different statistical populations. As a result, the student who navigated with the adaptivity modality presents, on average, an improvement in the domain knowledge of about $\Delta = 26\%$, being $\Delta = \overline{\Delta_X} - \overline{\Delta_Y}$, expressed in percentage with respect to $\overline{\Delta_Y}$. Moreover, applying the Hodges and Lehemann procedure [23, 24], we computed the estimator $\widehat{\Delta}$ of the $\overline{\Delta_S}$ variable: $\widehat{\Delta} = 2.6$. In other terms, we can say that students who use the system in the adaptive modality have an improvement in learning of about the 27.54% with respect to students who navigate in a *without* modality.

5.3 Layered Evaluation

The first evaluation has given positive results, but, as stated in several papers who addressed the evaluation of adaptive educational or not educational systems, it results a not trivial task to fully understand if the success or the failure of such an experimentation depends exclusively on the adaptive components [12, 8]. Other factors, e.g. usability factors [26], could have influenced the learning process. In this Subsection we propose the *Layered Evaluation* of the system, following the guidelines pointed out by Brusilovsky et al. in [8]. The main idea behind this approach to the evaluation is to decompose adaptation in two main distinct high-level processes: *Student Modelling* and *Adaptation Decision Making* and evaluating them separately. Moreover, this approach can facilitate re-use in context with different decision making modules [32].

5.3.1 Student Modelling Process

As we showed in Sec.3, our Student Modelling process is based on low-level information provided by the system during navigation, through a monitoring mechanism, based on the logging of some run-time student's actions. This evaluation aims to answer to the following question [8]:

Are the user characteristics being successfully detected by the system and stored in the user model?

In order to answer to this important question, the system, by means of its self-assessment form shown in Fig.8, asked the student to agree or not to agree with her virtual model, i.e., the student's current representation built by the system. The student could answer through a 7-point likert scale. In particular, the system did not explicitly ask for her LS or CS, which could be difficult to fully understand by a non-insider; on the contrary, the system exploited a simple non-technical language, fully comprehensible from high school students as well. Moreover, this process could be started by the student herself at run-time, by clicking a link in the navigation menu, i.e., *Guarda il tuo profilo (Check your model).* Involving students directly in the assessment of their own model is an important issue in order to evaluate the SM reliability [8, 21]. The system logged the frequency distribution illustrated in Fig.9. As we can see, most of the assessments, more than 90%, are located in the right part of the frequency distribution, i.e. most students agreed with their representation, strengthening the usefulness of this kind of SM representation and updating.

5.3.2 Adaptation Decision Making

In this Subsection, we evaluate some different aspects of the *Adaptation Decision Making* mechanism. The question is [8]:

Are the adaptation decisions valid and meaningful, for the given state of the student

Figure 8: The Self-Assessment Form.

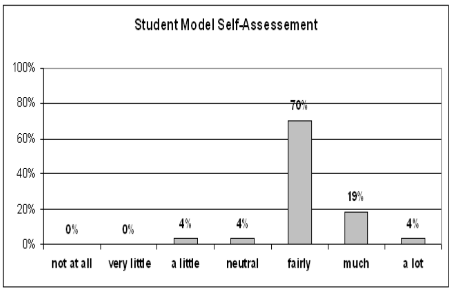

Figure 9: The self-assessment frequency distribution.

model?

The adaptive mechanism is based on the building of a new LOS on the basis of the SM and on the parameters illustrated in Sec.3, every time a student leaves a LN after having taken a post-LN questionnaire, in order to measure her knowledge about the KI belonging to that particular LN. The evaluation of this mechanism is decomposed in the following steps:

- Evaluating how much *students* agreed with the proposed new LOS.
- Evaluating how much *teachers* agreed with the proposed new LOS.

The system logged all the choices made by students. In Fig.10 the students' choices are shown. As we can see, the most important result is that the 60% of the Students followed the 100% of the suggested LOS while the 6.67% followed the 88.8% of the suggested LOS. More than 85% followed more than the 60% of the suggested LOS.

In [28] and in [29], we already presented an evaluation on the quality of the new LOS proposed by the system on the basis of the SM and of the other parameters needed by the adaptive algorithm to run. In [28] we presented a first evaluation based on two suggested LOS by the system. These two learning sequences were assessed by a sample of 14 teachers who were required to assess the instructional validity of the two proposed didactic plans compared to their related SMs. We submitted to the teachers the following question for both the *LOS*: *This Learning Object Sequence is a valid Learning Object Sequence on the basis of the starting student model SM?*, by means of a 5-points Likert scale (strongly disagree, disagree, neither agree nor disagree, agree, strongly agree). The experimental results showed that the 7.1% disagree, the 7.1% neither agree nor disagree, 71.4% agree and 14.4% strongly agree with the first didactic plan; 78.6% agree and 21.4% strongly agree for the second one. In the second paper [29] we addressed just this problem by presenting an extended experimentation through six case studies where six new *LOS* were proposed on the basis of the *SM* in the software programming domain. A sample of 30 teachers were asked to assess the didactic validity of two proposed plans. More than the 70% of teachers gave a positive assessment to the proposed *LOS*. The domain of this evaluation was the programming domain. This evaluation can be considered here also because of the independence of the mechanism from the particular domain.

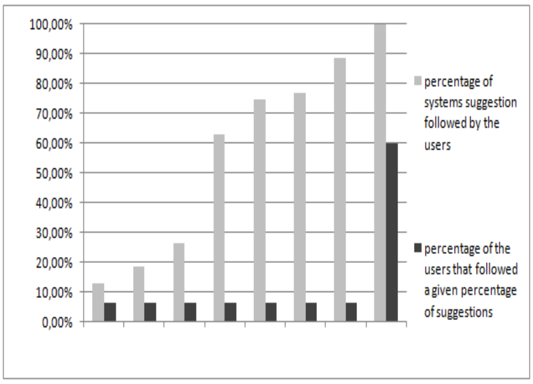

Figure 10: Analysis of the suggested *LOS* followed by students.

5.3.3 Discussion

The Layered Evaluation allowed us to decompose the adaptivity component into the student modelling component and the adaptation decision making. Both for the *SM* and *LOS* validation there are encouraging results: students showed to agree with the plans proposed by the system, while in another experimental plan, we showed that most teachers assessed them positively.

5.4 Navigation and Happy Sheet Analysis

Here we report some experimental data, gathered directly from students by means of an happy sheet submitted as a post-navigation questionnaire. This questionnaire was composed by 10 questions on the student's satisfaction degree in the use of the LECOMPS system. In Fig.11 we asked the enjoyment degree in the use of the system. As we can see, both adaptive and non-adaptive students enjoyed the system.

In Fig.12 the assessment of the graphical environment is shown.

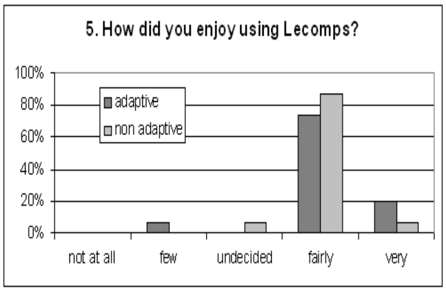

Figure 11: Student Satisfaction.

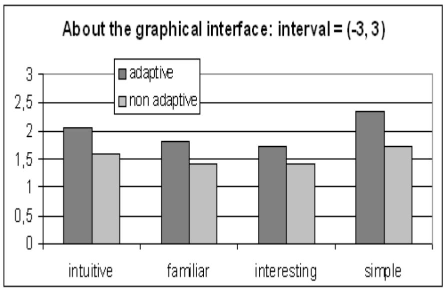

Figure 12: Assessment of the Graphical Environment.

Finally, in Tab. 4, some important navigation parameters are illustrated. In particular, we can see almost no significative differences between the two learning modalities in the number of visited nodes, time spent per-node and global navigation time. This is not in contradiction with the other results because, because, thanks to the WMW test, we can say that the quality of learning in the two modalities was however different.

6 Conclusion and Future Work

We described a system allowing for the management of personalization and adaptation of courses in e-learning. The LS-Plan framework selects and organizes the available learning objects so to adjust a learning path in accord with the initial learner cognitive state and learning styles. Then the path can be managed to allow for adaptation during course taking, basing on the modifications induced into the SM by the frameworks algorithms.

The cognitive state is updated in accord with the knowledge progresses of the learner. The learning styles are also updated during the course, depending on the results coming

Modality	Visited Nodes	Navigation Time	Time per Node
Non Adaptive	15	39	3
Adaptive	13	38	3

Table 4: Navigation Parameters

from assessment activities. Such updates can have consequences, during the taking of a personalized course, in both the learning path reconfiguration and the learners guidance, provided by the system by a step-by-step suggesting of the learning objects to take and proposing different contents if needed. The frameworks capabilities for course sequencing are provided by means of a planner, whose characteristics allow to achieve a sound basical behavior and a substantial enrichment on the expressivity of relations among concepts to be taught, providing help for the teacher in both configuring optimized courses and managing the pool of learning nodes. The goals in this last aspect are made possible to reach by the use of control knowledge, which is possible in planning domain description languages, and in PDDL-K in particular (the language used in the frameworks planner).

We think that LS-Plan capabilities to support both “static” personalized construction of a course and “dynamic” course maintainance and rebuilding and learners guidance, are an interesting hybridization of the main approaches to adaptation that are in our knowledge. Moreover, one main contribution of this paper is in the estensive experimentation we let the framework sustain. The evaluation is performed following mainly two standard procedures, the *As a Whole* and the *Layered* approaches, in order to prove the validity of the whole system and of the adaptive components separately, as suggested by the literature concerning the validation of adaptive systems. The results for both approaches are very promising. We exploited the implementation of the Lecomps system, embedding LS-Plan features in it and using it to deliver courses on *Italian Neorealist Cinema* to two distinct test groups (one provided with the full adaptive treatment of LS-Plan and the other one bravely getting a non-adaptive experience).

In terms of future work, we are designing future developments in the planner, aimed at extending the PDDL-K with additional syntactic elements to provide the teacher with powerful control knowledge means for the management of learning nodes in pools and educational environments.

References

[1] E. Alfonseca, R. M. Carro, E. Martin, A. Ortigosa, P. Paredes, ”The impact of learning styles on student grouping for collaborative learning: A case study”, User Modeling and User-Adapted Interaction, vol. 16, no. 3-4, pp.377401, 2006.

[2] N. Bajraktarevic, W. Hall, and P. Fullick, ”Incorporating learning styles in hypermedia environment: Empirical evaluation”, Proc. Fourteenth Conference on Hypertext and Hypermedia, pp. 41-52, 2003.

[3] M. Baldoni, C. Baroglio, V. Patti, L. Torasso, ”Reasoning about learning object metadata for adapting SCORM courseware”, Proc. of International Workshop on Engineering the Adaptive Web: Methods and Technologies for personalization and Adaptation (EAW’04), 2004.

[4] M. Baldoni, C. Baroglio, I. Brunkhorst, E. Marengo, V. Patti, "Reasoning-Based Curriculum Sequencing and Validation: Integration in a Service-Oriented Architecture", EC-TEL 2007, LNCS 4753, pp. 426-431, 2007.

[5] B.S. Bloom, "Taxonomy of Educational Objectives", David McKay Comp. Inc., 1964.

[6] P. Brusilowsky, "Adaptive hypermedia", User Modeling and User-Adapted Interaction, vol. 11, pp. 87110, 2001.

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

[8] P. Brusilovsky, C. Karagiannidis, and D. Sampson, "Layered evaluation of adaptive learning systems", International Journal of Continuing Engineering Education and Lifelong Learning, vol. 14, no. 4/5, pp. 402-421, 2004.

[9] P. Brusilovsky and E. Millan, "User models for adaptive hypermedia and adaptive educational systems", In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, The Adaptive Web: Methods and Strategies of Web Personalization, vol. 4321 of Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg New York, 2007.

[10] T. Bylander, "The computational; complexity of propositional strips planning", Art. Intell. 69, 1-2, pp.165-204, 1994.

[11] C. A. Carver, R. A. Howard, and W. D. Lane, "Enhancing student learning through hypermedia courseware and incorporation of student learning styles", IEEE Trans. on Education, vol. 42, no. 1, pp.33-38, 1999.

[12] D. N. Chin, "Empirical evaluation of user models and user-adapted systems, "User Modeling and User-Adapted Interaction", vol. 11, no. 1, pp. 181–194, March 2001.

[13] 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., vol. 1, no. 16, pp. 63-89, Jan 2007.

[14] A. Cristea, A. De Mooij, "LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators", Proc. of WWW03, Alternate Education track, 2003.

[15] A. Cristea, L. Calvi, "The three Layers of Adaptation Granularity", Proc. of UM 2003, LNAI 2702, pp. 414, 2003.

[16] A. Cristea, A. De Mooij, "Adaptive Course Authoring: My Online Teacher", Proc. of ICT03 vol. 2, pp. 1762-1769, 2003.

[17] P. De Bra, D. Smits, and N. Stash, "Creating and delivering adaptive courses with AHA!", Proc. EC-TEL '06, pp. 21-33, 2006.

[18] J.C. Falmagne, M. Koppen, M. Villano, J.P. Doignon, and L. Johannesen, ”Introduction to knowledge spaces: How to build, test, and search them”, Psychological Review, vol. 97, no. 2, pp. 201-224, April 1990.

[19] R. M. Felder and L. K. Silverman, ”Learning and teaching styles in engineering education”, Engineering Education, vol. 78, no.7, 674-681, 1988.

[20] R. M. Felder and J. Spurlin, ”Application, reliability and validity of the index of learning styles”, Int. Journal of Engineering Education, vol. 21, no. 1, pp. 103-112, 2005.

[21] C. Gena, ”Methods and techniques for the evaluation of user-adaptive systems”, Knowl. Eng. Rev., vol. 20, no. 1, pp. 1-37, 2005.

[22] S. Graf, Kinshuk, ”Providing adaptive courses in learning management systems with respect to learning styles”. Proc. of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (eLearn 2007), pp. 25762583, 2007.

[23] J. Hodges and E. Lehmann, “The efficiency of some non parametric competitors of the t-test”, Ann. Math. Statist, vol. 35, pp. 324–335, 1956.

[24] J. Hodges and E. Lehmann, “Estimates of location based on rank tests”, Ann. Math. Statist, vol. 34, pp. 598–611, 1963.

[25] M. Hollander, D. Wolfe, and R. Cohen, ”Nonparametric Statistical Methods”, John Wiley and Sons, 1973.

[26] K. Höök, ”Evaluating the utility and usability of an adaptive hypermedia system”, Proc. 2nd international conference on Intelligent user interfaces (IUI’97), pp. 179-186, 1997, doi:10.1145/238218.238320.

[27] E. Sangineto, N. Capuano, M. Gaeta, A. Micarelli, ”Adaptive course generation through learning styles representation”, Universal Access in the Information Society (UAIS), vol. 7, no. 1-2, pp. 1-23, 2008.

[28] C. Limongelli, F. Sciarrone, and G. Vaste, “Ls-plan: An effective combination of dynamic courseware generation and learning styles in web-based education”, Proc. of 5-th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2008), vol. 5149 of Lecture Notes in Computer Science, pp. 133-142, Springer Berlin/Heidelberg, 2008.

[29] C. Limongelli, F. Sciarrone and G. Vaste, ”An Application of the LS-Plan system to an educational hypermedia”, Int. J. of Web-Based Learning and Teaching Technologies, 4, 1, pp. 16-34, 2009.

[30] Learning Technology Standards Committee. Draft Standard for Learning Object Metadata, IEEE 1484.12.1-2002, 2002.

[31] J. Masthoff, "The evaluation of adaptive systems", Hershey, PA, USA: IGI Publishing, 2003, pp. 329-347.

[32] N. Manouselis and D. Sampson, "Agent-based e-Learning Course Discovery and Recommendation: Matching Learner Characteristics with Content Attributes", International Journal of Computers and Applications (IJCA), Special Issue on Intelligence and Technology in Educational Applications, 25(1), 50-64, July 2003

[33] P. Mohan, J. Greer, and G. McCalla, "Instructional planning with learning objects", Proc. IJCAI-03 Workshop 'Knowledge Representation and Automated Reasoning for E-Learning Systems, Kohlhase Melis Baumgartner, Cairns, editor, 2003.

[34] J. Piaget, "Language and thought of the child", New York: Harcourt, 1926.

[35] S. Siegel and N. J. Castellan, "Nonparametric statistics for the behavioral sciences", New York, USA: McGraw-Hill Book Company, 1988.

[36] A. Sterbini, and M. Temperini, "A logical framework for course configuration in e-learning", Proc. 4th IEEE Int. Conf. on Information Technology Based Higher Education (ITHET03), July 7-9, Marrakesh, Morocco, 2003.

[37] M. Temperini, U. Vitti, "A Web Application for Automated Course Configuration", Proc. 8th International Conference on Information Technology Based Higher Education and Training (ITHET2007), 10th-13th July 2007, Kumamoto, Japan, 2007

[38] G. Weber and P. Brusilovsky, "Elm-art: An adaptive versatile system for web-based instruction", International Journal of Artificial Intelligence in Education, vol. 12, pp. 351-384, 2001.

[39] F. Wilcoxon, "Probability tables for individual comparisons by ranking methods, "Biometrics", 3, pp. 119–122, 1947.

[40] H. Wu, "A Reference Architecture for Adaptive Hypermedia Applications", PhD thesis, Eindhoven University of Technology, The Netherlands, ISBN 90-386-0572-2, 2002.

[41] M. S. Zywno, J. K. Waalen, "The effect of hypermedia instruction on achievement and attitudes of students with different learning styles". Proc. Annual ASEE Conf, 2001.