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

Carla Limongelli, Filippo Sciarrone, Marco Temperini, and Giulia Vaste

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 ensuring the maintenance of such courses, for continuous adaptation of the learning material proposed to the learner. Adaptation is meant, in this case, with respect to the knowledge possessed by the learner and her learning styles, both evaluated prior to the course and maintained while attending the course. 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 based on the initial learner’s Cognitive State and Learning Styles, as assessed through pre-navigation tests. During the student’s 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 on an extensive experimental evaluation, performed by integrating LS-Plan in an educational hypermedia, the LECOMP S 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 by mainly following two standard procedures: the As a Whole and the Layered approaches. The results are encouraging both for the system on the whole and for the adaptive components.

Index Terms—Personalized learning, adaptive web-based system, learning styles.

1 INTRODUCTION

Modern research in hypermedia systems focuses mostly 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 modeling and adaptation to meet current user needs. Consequently, it requires some standard experimental procedures ensuring the evaluation of adaptive systems so as to evaluate the added value given by the adaptive components and processes. Over the past years some authors have addressed this issue to give researchers useful guidelines for the evaluation of adaptive systems, as in the work of Chin [12], Brusilovsky et al. [7], Gena [23], and Masthoff [34]. In this context, personalization and adaptation in educational systems are often associated with Course Sequencing, which produces an individualized sequence of didactic materials or activities for each student, dynamically selecting the most appropriate ones at any moment. In this context, a widely used approach is Dynamic Courseware Generation [9], where the personalized course is generated so as to guide the learner starting from her initial state of knowledge, allowing her to cover a given set of learning goals and eventually ensuring that the course content is adapted to the learner’s progress.

In this paper, we propose an extended evaluation of the LS-PLAN system, a system capable of providing educational hypermedia with adaptation and personalization on the basis of the student’s knowledge and learning styles [30]. Unlike the other adaptive educational hypermedia, LS-PLAN provides a personalization engine that can be plugged in any educational system.

Two main features of the LS-PLAN system concern the learning styles management and the kind of sequencing generated. Adhering to the Felder and Silverman’s Learning Styles Model [21], LS-PLAN models learning styles as tendencies and estimates how the didactic material affects the success of the learning activity. In particular, the teacher associates, with the learning nodes, some weights (associated to learning styles) that represent the suitability of that material for learning preferences. If the student studies a given material with success, it means that the presentation style is consistent with the student’s way of learning, so the student’s learning styles move toward the learning styles of the node; on the contrary, if the study does not succeed as it should, the student’s learning styles move in the opposite direction of the learning styles of the node. LS-PLAN provides sequencing by planning the whole learning path ahead of the course and rebuilding it, possibly step by step, while taking the course.

For our evaluation, we embedded LS-PLAN into the LECOMP S learning system [39], and fed it with an educational environment on Italian Neorealist Cinema. We investigate the following main research question:

Does the LS-PLAN Adaptivity Mechanism give added value to learners?
To answer this research question, we developed an experimental plan. First, we managed to involve a set of 30 individuals in the experiment, by means of a standard sample selection procedure. We then performed two main empirical evaluations: the classic As a Whole evaluation and the Layered evaluation, following the guidelines proposed in the literature of experimental evaluations of adaptive systems [7], [12], [23]. In the first evaluation, participants were partitioned into two groups: one using the basic hypermedia, i.e., without the adaptive features provided by the LS-PLAN engine, and the other using the LECOMPS-LS-PLAN integrated system. The Layered evaluation aimed to check the adaptive components separately. In particular, we evaluated separately the aspect of user modeling and of adaptation decision making. The result of the As a Whole evaluation was an added value equivalent to a 27.54 percent increase in knowledge for the students who navigated in the With modality versus students who navigate in the Without modality. In the Layered evaluation, we obtained positive results both for the student model representation and for the Adaptation Decision Making. Finally, we present other statistics concerning the navigation parameters as logged by the system, a questionnaire on user attitudes and affect analysis, and a quick look at a case study for the dynamical evolution of learning styles. These statistics have given good indications too.

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 Section 6, our conclusions are drawn.

2 RELATED WORK

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., Dynamic Courseware Generation (DCG) [9], the work of Baldoni et al. [4], [3], and the IWT system [37].
- 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 [20] and the ELM-ART system [40]).

In this section, we report on the work related to these two approaches to sequencing and on methodologies for student knowledge modeling and learning styles management.

2.1 Sequencing/Resequencing

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 [4], [3], in which learning resources (learning objects in [4] or courses in [3]) 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 [15]. The definition of these metadata is based on ontologies of interest, to guarantee shared meanings, interoperability, and reusability, ensuring a Semantic Web perspective. However in these approaches, “tagging” is a bottleneck: teachers may find it hard to adhere to predefined ontologies. Moreover, in [4], [3], personalization is not performed at when it comes to 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 [9] that creates a plan of the course contents, follows the student during the fruition of the course, and makes a replanning if the student fails to demonstrate the acquisition of a concept. Sequencing in DCG is sophisticated, and considers some personal characteristics, although—to our knowledge—it does not let resequencing actions depend on the occurring learning styles modifications.

2.2 Implicit/Step by Step

Adaptive navigation support techniques are widely used in AHA! [20] and in ELM-ART [40]. 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 [43] 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 [17] allows authoring based on LAOS [18] and LAG [16] models, making it possible to use an adaptation language to program adaptive behaviors, which will be compiled in suitable rules. However, authors are required either to possess programming skills or to rely upon predefined strategies. Moreover, AHA! has not exploited 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 is more granular from a cognitive state viewpoint, as shown in the next section, but it does not consider learning styles.

2.3 Student’s Knowledge

The overlay approach is the most used one for modeling user knowledge [8]. It can use Boolean, qualitative, or quantitative values for indicating if and how much a fragment of the domain is thought to be already known by the student; it can be layered for taking into consideration the different sources used for the estimations of the user knowledge. Moreover, the overlay approach can model conceptual or procedural knowledge, and it can be expanded through a model for taking into consideration user misconceptions. Bug models are especially used for procedural knowledge; their practical use is complicated and is limited to Intelligent Tutoring Systems based on
simple domains. LS-PLAN uses a qualitative overlay model, using three levels of Bloom’s Taxonomy. The student knowledge is estimated on the basis of tests, and if they are not present, it is estimated by considering the “pages-seen.” LS-PLAN, differently from ELM-ART, does not provide a bug model: it is hard to model user misconceptions in a wide and nonpredefined domain. Some adaptive hypermedia use an approach for student knowledge management similar to the LS-PLAN one, and in some respects more advanced. AHA! does not exploit assessment for student model updating, and it is based only on the user browsing behavior. However, it provides an interesting mechanism of knowledge propagation, that is, modifying the estimate of the knowledge of a given concept on the basis of the estimate of the knowledge of a related concept. AHA! allows the authors to define different relationships among concepts and the correspondent knowledge propagation mechanism [19]. Netcoach [41], developed on the basis of the latest version of ELM-ART [40], uses a layered overlay model, composed by pages visited by the student, tests, inferences about the knowledge of a concept on the basis of the student’s success in more advanced ones, and concepts marked as known by the student. Netcoach builds a fifth layer, the learned layer, on the basis of the other levels, i.e., a concept is assumed learned if it is either tested, inferred, marked, or, in the absence of tests, visited. The LS-PLAN approach is currently less granular, and it is heavily based on tests: browsing a material or acquiring a more advanced concept is not considered sufficient (if tests are available) for estimating a known concept. TANGOW [1] allows to store information on the actions the student performed while interacting with the system, including exercises scores and visited pages. Moreover, it provides a formalism that allows the course author to specify the necessary adaptation rules. So, TANGOW on the one side is a flexible system, but on the other side leaves critical responsibilities to the course author. In [4] and [3], a management of student knowledge very similar to the LS-PLAN one is proposed, at least in the phase of course construction. However, the authors assume that the user’s competencies can only increase during the study, without considering “forgetfulness.” LS-PLAN, through its adaptation algorithm, allows to estimate the presence of a lapse of memory or of a wrong estimation of student knowledge about a given concept.

2.4 Learning Styles
The actual effectiveness of learning-styles-based adaptation is still a matter for discussion: It is questioned and supported, as illustrated in [8]. An empirical evaluation in [5] shows no relevant improvement in the attainment of primary schools students using an adaptive learning-styles-based hypermedia with respect to their colleagues treated more traditionally (in particular, the learning styles were modeled through the sequential-global dimension of Felder and Silverman’s Learning Styles Model [21]).

On the other hand, many studies have been conducted applying the idea that teaching strategies, based also on student learning styles, might increase the learner’s motivation, comprehension, participation, and learning effectiveness. In particular, Felder and Silverman’s Learning Styles Model has been often taken into consideration in the literature ([37], [24], [2], [11], [1]). The reason for such attention appears to be manifold: 1) this model is a combination of other models, such as Kolb’s and Pask’s ones [29], [35]; 2) it provides a numerical evaluation of learning styles, which is a useful factor in computer-based systems; and 3) its reliability and validity has been successfully tested, such as in [32], [44].

Coffield et al. [14] state that “Different theorists make different claims for the degree of stability within their model of styles”; following this problem, Felder and Spurlin in [22] state that learning styles are tendencies and they may change during the educational experiences; this claim has been also empirically shown in [5].

The Felder and Silverman’s model is used in a lot of systems, such as the following:

- the add-on for the Moodle Learning Management System proposed in [24], 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, which initializes the student model in an explicit way through the Felder and Solomon’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, and
- the CS383 system [11] and the IWT system [37] 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—and 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 modifies it if necessary.

3 The Adaptive System
Fig. 1 shows the overall system. The LS-PLAN system provides the educational hypermedia with adaptivity; the main components are highlighted with gray 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 to evaluate the student’s starting knowledge, 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., a test, developed by Felder and Soloman (available at http://www.engr.ncsu.edu/learningstyles/ilsweb.html), which extracts the student’s learning preferences according to the four dimensions of the Felder and Silverman Model: active-reflective, sensing-intuitive, visual-verbal, sequential-global [21]. 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 also specifies her didactic strategies and defines her own instructional goal for each student. This information, together with both the results of the two initial questionnaires and the descriptions of the learning nodes, i.e., the Domain Knowledge, is coded in PDDL (see Section 3.3) files and sent to the Pdk Planner.

The Pdk Planner produces in output to the hypermedia a personalized Learning Object Sequence (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 studying each learning node. This information is used both for updating the student model and for the adaptation decision making, as is discussed in Section 3.1.2.

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

**Definition 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 $KI_\ell$, with $\ell \in \{K, A, E\}$, represents a cognitive level taken from Bloom’s Taxonomy: Knowledge, Application, and Evaluation.

We have chosen only three out of the six levels of Bloom’s taxonomy in the cognitive area, in order to test the correct behavior of the planner: it is easy, but heavy, to provide the $KI$ with all the six levels.

**Definition 2: (LEARNING STYLE).** A Learning Style $LS$ is a 4-tuple: $LS = (D_1, D_2, D_3, D_4)$, with $D_i \in [-11, +11]$, $i = 1, \ldots, 4$, where each $D_i$ is a Felder and Silverman Learning Style Dimension, i.e., $D_1$: active-reflective, $D_2$: sensing-intuitive, $D_3$: visual-verbal, and $D_4$: sequential-global.

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

**Definition 3: (LEARNING NODE).** A Learning Node $LN$ is a 5-tuple: $LN = (LM, AK, RK, LS, T)$, where

- $LM$ is the Learning Material, i.e., any instructional digital resource.
- $AK$ is the Acquired Knowledge. It is a $KI_\ell$ that represents the knowledge that the student acquires at a given level as specified in Definition 1, after having passed the assessment test related to the $KI_\ell$ of the node. If such a test is not present in the node, then the $AK$ is considered acquired anyway.
- $RK$ is the Required Knowledge. It is the set of $KI_\ell$ 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 2.
- $T$ is a pair of reals $T = (t_{min}, t_{max})$ that represents the estimated time interval for studying the material of the node, as prefixed by the teacher. Obviously, it is not possible to know if the fruition time ($t_f$) is actually spent on studying or if it is affected by other factors. However, the thresholds $t_{min}$ and $t_{max}$ that we consider, allow to eliminate at least two student
behaviors: the so-called “coffee break” effect, when the fruition time \( t_f \) is greater than \( t_{\text{max}} \), and a casual browsing of a given material, when \( t_f \) is less than \( t_{\text{min}} \).

**Definition 4:** (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 5:** (DOMAIN KNOWLEDGE). The Domain Knowledge DK is the set of all the KI present in a pool.

**Definition 6:** (COGNITIVE STATE). The Cognitive State CS is the set of all the KI possessed by the student with respect to the given topic: \( CS \subseteq DK \).

**Definition 7:** (STUDENT MODEL). The student model SM is a pair: \( SM = (CS, LS) \), where CS is given in Definition 6 and LS is given in Definition 2.

**Definition 8:** (TEST). A Test is a set of \( k \) items, i.e., questions, with \( k \in \mathbb{N} \). Each item is associated with a weight \( Q_j \in \mathbb{R} \). Each item has \( m \) answers, with \( m \in \mathbb{N} - \{0, 1\} \), and each answer is associated with a weight \( w_j \in \mathbb{R} \). \( S_{KI} \) is the score associated with a test: it assesses the student knowledge of the single KI.

Let us point out that in our system, questions are currently related to the acquirement of a given knowledge item. Including questions into learning nodes treating such topics allows us to “contextualize” the questions. However, the separation of the test from the learning nodes, ensuring the association of a given question with more than one knowledge item, can be feasible.

3.1 The Adaptation Engine

In this section, we show the mechanisms of the student model 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 consisting of some questions (see Definition 8), related to the knowledge items of the Domain. All the acquired knowledge items initialize the cognitive state \( CS \), 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 learning styles.

3.1.2 Student Model Updating and Adaptation Methodology

A revised version of the student model updating and adaptive decision-making algorithms presented in [30] has been proposed in [31]. 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: 1) update of the student model and 2) computation of the Next Node to be proposed, together with the new Learning Object Sequence. Basically, the idea is to work as the teacher would do: reexplaining the failing concept (proposing the same learning material as before), then trying to propose different learning material for the same concept, and finally on further fail, assuming that some of the prerequisites, previously taken for granted, are the source of the problem and will be suggested for rechecking.

Figs. 2 and 3 present the algorithms related to these two actions, respectively.

When the student studies an LN, the function UPDATESM is activated taking in input the LN and the current SM. The function TIMESPENTONTHENODE computes and returns the time \( t_f \), that is, the time the student spent on the node. The function COMPUTESCOREPOSTEST computes and returns the score taken by the student in the posttest related to the KI of the node, namely, the AK related to the LN. Should the posttest not be provided, we can only assume a score of 0 and consider the KI related to that node as acquired, though without updating the student’s LS. On the other hand, if the posttest is available, we update the student’s LS according to the LS associated with...
the node, to the fruition time $t_f$, and to the score obtained with the posttest. In particular, if the $KI$, is acquired, we consider that learning material is adequate for the case, so we can update the student’s $LS$ toward the $LS$ of the node, by an extent depending directly on score and inversely on time $t_f$. On the contrary, if the knowledge item is not acquired, we apply the opposite behavior. Let us note that such modifications are to be considered “adjustments” for the present $LS$ estimate, so they are actually quantified as values in $[0,1]$. The two functions $\eta_1$ and $\eta_2$ range in $[0,1]$, as shown in [31].

The function NEXTNODE proposes the next node to be learned on the basis of the new student model, as described in Fig. 3. If a $D_i$, as given in Definition 2, changes sign, we reckon there is a significant variation in the student $LS$, which makes it necessary to replan 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. Should that be the 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 an alternative node, $LN’$, having the same $RK$ and the same $AK$ of the current $LN$, and having the smallest distance from the student’s $LS$ (in terms of euclidean distance metric). If such a node does not exist, by means of the function ORDEREDPREDECESSORS 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) first, the predecessor nodes that were not yet visited; 2) then the nodes that do not provide tests, proposed according to the Difficulty levels $K,A,E$; 3) then the nodes that provide tests 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 replans a new $LOS$ and proposes its first $LN$. 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.

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] (available at http://pdk.dia.uniroma3.it). 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. The following is an outline of the Pdk functionalities that are exploited by LS-PLAN; for a more detailed description see [31]:

1. Course sequencing: The initial conditions are given by the cognitive state of the initial student model. The goal of the problem corresponds to course target knowledge. In this way, course sequencing is the synthesis of the actions that the planner produces to reach the goal.
2. Domain validation: This checks pool consistency, to spot actions that can never be executed, in the style of [20]. The loop check is an easy control for the planner: Starting knowledge is empty and target knowledge is the set of all knowledge items.
3. Redundant formula detection: This phase can help the teacher in arranging the pool of learning nodes.
4. 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 Definition 1), the particular way the teacher prefers to explain a given concept, or the constraint about the execution of some actions the teacher believes are mandatory for all the students, even if they demonstrate they 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 postconditions of the actions and the number of goals [10]. Moreover, the definition of correct control knowledge is also a difficult task and can generate inconsistency problems. However, the high-level control formulas provided by PDDL-K give a set of predefined schemata and allows one to easily and naturally specify heuristic knowledge, without requiring specific programming skills. An appropriate heuristic knowledge can prune the search space and 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] show that pools with up to 100 nodes can be managed by the planner in less than 5 seconds.
As mentioned 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.

Since a full description of the system is not in the focus of this paper, we will very briefly address it in the following three sections, concluding by introducing the experimental evaluation described in Section 5.

4.1 Educational Environment, Enrolment, and Course Delivery

A prospective learner can see the information related to the educational environment, enroll in it, and submit the questionnaires to input her initial cognitive state in the system, as related to the subject matter and evaluation of learning styles.

When the personalized course is available (see next section) the learner can access and take the learning material. Two examples of access page to a course are shown later on, in Figs. 4 and 5.

4.2 Course Construction

Once the initial cognitive state and learning styles are available to a learner, it is possible to activate the process of automated configuration of the course via the LS-PLAN. In the present version of the system, we preferred not to let such process start automatically; instead, it is activated by the teacher through a suitable interface, where the initial cognitive state of the learner and the aimed target knowledge are shown and can possibly be modified.

The experimental evaluation described in the next section is based on the use of two different versions of LECOMPS: one enabling the full application of the LS-PLAN framework and another one providing a nonadaptive management of courses.

5 Evaluation

In this section, we show an extended empirical evaluation of the LS-PLAN system by experimenting its embedding into the LECOMPS hypermedia. This evaluation completes the very first experimentation presented in [30] and the experimental results obtained in [31], where simulated case studies have been addressed only for the analysis of the adaptive algorithm behavior. In the case in point, our main research question concerns the reliability and the added value given by the adaptive framework, once it is applied to
needed (contact authors for login). The server is a protected server, and a signing procedure is large-scale experiments in humanistic fields too, in the designer. We chose this domain in order to be able to run Neorealist Cinema.

In Section 5.2, we propose the parameters needed to start the experiments are drawn. In this section, we show the experimental environment setup, where all the following four questionnaires were built and proposed:

5.1 Experimental Setup

In this section, we show the experimental environment we built to run our experiments, available at http://paganini.dia.uniroma3.it website, together with all the raw data and questionnaires. It runs on a Linux departmental server and is based on a java application, for the LS-PLAN system, communicating with the php-based LECOMPS hypermedia. The server is a protected server, and a signing procedure is needed (contact authors for login).

5.1.1 The Knowledge Domain

We used the LECOMPS hypermedia to teach topics on Italian Neorealist Cinema. A film critic has also been involved in the project as domain expert, together with an instructional designer. We chose this domain in order to be able to run large-scale experiments in humanistic fields too, in the future. We built a knowledge domain consisting of 18 LN, each one having an associated test, and of 12 K1, selected by the domain expert.

5.1.2 Questionnaires

To evaluate the student’s knowledge and satisfaction, the following four questionnaires were built and proposed:

- **Pretest questionnaire:** A questionnaire formed by 50 items to check the starting knowledge of the sample. It was built to measure the starting knowledge of each participant on Italian Neorealist Cinema. It was designed as a set of multiple choice and true/false items. A domain expert and an instructional designer helped us build each item. An item example is: “Was Roberto Rossellini born in a poor family?”

- **Posttest questionnaire:** The posttest questionnaire, consisting of 50 items, aimed to measure the acquired knowledge after having surfed the system, in order to verify if and how much the system itself was useful to help learners reach the didactic goals. Similarly to the pretest questionnaire, this one was designed as a multiple choice and true/false questionnaire, with the support of a domain expert and an instructional designer.

- **LN questionnaire:** A questionnaire for every LN was prepared to evaluate the acquired knowledge after having studied it. Even this test was designed by means of true/false and multiple choices items.

- **Questionnaire on user attitudes and affect:** This is a questionnaire formed by 13 items designed to measure the students’ ratings on some navigational and usability aspects. This questionnaire encompasses ten 5-point Likert scale items, two Semantic Differentials, and one Fill-in item. An item example is: “How difficult did you find your task?” The questionnaire on user attitudes and affect was submitted at the end of the course, as the last step of the experimentation. We followed the guidelines expressed in [36] for the design of such a questionnaire.

5.1.3 The Sample

The sample was randomly selected among students from Universities, students from high schools, teachers, and people who were interested in learning something about Italian Neorealist Cinema. The process of sample gathering has been divided into several steps. In the first step, we selected 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 submitted to the whole group a questionnaire containing items about the most important issues addressed by the learning domain. In the third step, we formed a homogeneous group of 30 individuals out of the initial 45 with the lowest average, i.e., the lowest starting knowledge on the domain and the lowest possible dispersion around it.

We obtained one group of individuals with average \( \bar{x} = 6.81 \) and standard deviation \( S_x = 4.36 \). We considered these two values a good compromise between a low starting knowledge and an acceptable dispersion, to minimize the well-known statistical problem due to the between subjects dispersion [12], [34]. In this way, we reduced the sample to 30 users, equally distributed between males and females whose age fell within the range [20, 50]. In particular, on average, our sample started with a domain knowledge of 16.23 percent, i.e., every individual obtained, on average, the 16.23 percent of the maximum possible score.

In Fig. 6, the sample LS distribution is shown. In particular, we have the same distribution in the Active-Reflective dimension for both experimental group \( X \) and control group \( Y \). In the other three dimensions, the sample appears almost homogeneous.

5.2 The As a Whole Evaluation

In this section, we show the controlled experiment in the With versus Without adaptive component, i.e., the LS-PLAN engine, to investigate our research question. The Without version of the overall system was composed by the LECOMPS system only. Students were free to navigate and to reach their didactic goals without any sort of guidance. The With version was the complete system, i.e., LS-PLAN plus LECOMPS.
5.2.1 The Research Question

The research question (RQ1) is:

\[ RQ_1: \text{Do students navigating with the adaptive modality learn more than students navigating without the adaptive modality?} \]

5.2.2 The Statistical Model

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

- **Independent variable:** We defined the independent variable \( \Delta_S \) as the difference between the score obtained by each student in the postnavigation test \( S_{\text{post}} \) and the one obtained in the prenavigation test, \( S_{\text{pre}}: \Delta_S = S_{\text{post}} - S_{\text{pre}} \). 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 \( \Delta_S \) belongs to, is the normal distribution (e.g., [38], [27]). This assumption strengthens the experiment because it follows a more general statistical approach.

- **Use of the Test of Wilcoxon-Mann-Whitney (WMW) for two independent samples:** This test, which proceeds from the psychological research area, is well suited for testing in experiments where humans play a crucial role [42] and where one has to verify a simple shift toward higher values of the median \( \theta \) of a stochastic independent variable. Besides, this test is a powerful one and corresponds to the parametric \( t \)-test. 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 \( \Delta_X \) and \( \Delta_Y \) the values of the variable \( \Delta_S \), respectively, for the group \( X \) and \( Y \).

5.2.3 Data Gathering

Students of both groups were required to navigate into the system for 45 min. at the most. We gathered all the pretest scores \( S_{\text{pre}} \) and all the posttest scores \( S_{\text{post}} \) to compute the variable \( \Delta_S \). Table 1 shows the main statistical parameters. On average, students who navigated into the adaptive environment showed a better improvement than the students who navigated into the nonadaptive environment. Moreover, the Standard Deviation \( S_{\Delta_S} \) is the same for both groups.

<table>
<thead>
<tr>
<th>Modality</th>
<th>St. Dev. ( S_{\Delta_S} )</th>
<th>Mean ( \Delta_S )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive</td>
<td>4.06</td>
<td>10.30</td>
</tr>
<tr>
<td>Non-Adaptive</td>
<td>4.29</td>
<td>7.61</td>
</tr>
</tbody>
</table>

5.2.4 Hypothesis Testing

Here, we show the WMW testing procedure applied to our statistical data. First, we define the Null Hypothesis \( H_0 \): the two learning modalities show a nonsignificant statistical difference between them; the variables \( \Delta_X \) and \( \Delta_Y \) belong to the same statistical distribution. Second, we define the Alternative Hypothesis \( H_1 \): The two learning modalities show a significant statistical difference between them; the variables \( \Delta_X \) and \( \Delta_Y \) belong to different statistical distributions, i.e., we have \( \theta_{\Delta_X} > \theta_{\Delta_Y} \), being \( \theta_{\Delta_X} \) and \( \theta_{\Delta_Y} \) the medians of the two statistical distributions, respectively, of \( \Delta_Y \) and \( \Delta_X \). This would mean that the statistical distribution of the adaptive modality is shifted toward higher values of acquired knowledge. Third, we fixed our significance level \( \alpha = 0.05 \). Finally, following the standard WMW procedure, we obtained a \( p - value = 0.03 < \alpha \). The Null Hypothesis \( H_0 \) can be rejected and the alternative Hypothesis \( H_1 \) can be accepted.

5.2.5 Between Groups Analysis

We performed the analysis of the statistical differences between groups for the As a Whole evaluation, by means of the nonparametric two-tails U-Test [38] with its associated power analysis, as suggested in [12].

- Null Hypothesis \( H_0 \): There is no difference between the experimental group \( X \) and the control group \( Y \): the two groups began with the same starting knowledge on Italian Neorealist Cinema.
- Alternative Hypothesis \( H_1 \): The two groups \( X \) and \( Y \) are different in terms of starting knowledge on Italian Neorealist Cinema.
- Significance Level \( \alpha = 0.05 \).

We obtained \( p - value = 0.25 \) and \( power = 0.732 \). \( H_0 \) can be accepted strengthening the nondifference hypothesis between \( X \) and \( Y \) while the power value is close to the value suggested in [12], that is, \( power = 0.8 \). We accept this value as a good compromise between the number of participants and the probability of 73 percent to reject \( H_0 \) when false.

5.2.6 Discussion

Here, we point out the statistical results of the On the Whole evaluation. This evaluation showed that the two independent variables \( \Delta_X \) and \( \Delta_Y \) 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 \( \Delta = 26 \) percent, being \( \Delta = \Delta_X - \Delta_Y \), expressed in percentage with respect to \( \Delta_Y \). Moreover, applying the Hodges and Lehmann procedure [25], [26], we computed the estimator \( \hat{\Delta} \) of the \( \Delta_S \) variable: \( \Delta = 2.6 \). In other terms, we can say that students who used the system in the adaptive modality experienced an improvement in learning of about 27.54 percent as
opposed to students who navigated in a Without modality. Finally, the U-Test with its associated power analysis strengthen the replaceability of these results.

5.3 Layered Evaluation

The As a Whole evaluation has given positive results, but, as stated in several papers that addressed the evaluation of adaptive educational or noneducational systems, it is not a trivial task to fully understand whether the success or the failure of such an experimentation depends exclusively on the adaptive components [12], [7]. Other factors, e.g., usability factors [28], might have influenced the learning process. In this section, we propose the Layered Evaluation of the system, following the guidelines pointed out by Brusilovsky et al. in [7]. The main idea behind this approach is to decompose adaptation into two main distinct high-level processes: Student Modeling and Adaptation Decision Making, and evaluate them separately. Moreover, this approach can facilitate reuse with different decision-making modules [33]. This evaluation allows us to better verify the contribution of the system adaptivity separately.

5.3.1 Student Modeling Process

As we showed in Section 3, our student modeling process is based on low-level information provided by the system during navigation, through a monitoring mechanism based on the logging of some student’s actions. This evaluation aims to answer to the following research question RQ2:

RQ2: Are the user characteristics being successfully detected by the system and stored in the user model?

In order to answer RQ2, during navigation, the system provides an assessment form, allowing the learner to express agreement or disagreement with her current virtual model. In this form, the learner is shown the current representation of her student model and is asked to declare her agreement or disagreement through the 7-point Likert scale of Fig. 7. The language used to show the student model tries to be nontechnical and fully comprehensible for noninsiders. The form is available at any time, while attending the course, through a suitable link included in the navigation menu. Involving students directly in the assessment of their own model is an important issue in order to evaluate the student model reliability [7], [23]. The system logged the frequency distribution illustrated in Fig. 7. The distribution of the students’ rating is unbalanced toward positive values, i.e., the 93 percent is on the right side of the distribution. We think that most students, who navigated in the adaptive environment, deemed their model “fairly accurate” because of the short amount of time spent surfing and because of the small number of available nodes. In fact, the student model, being a dynamic representation of the student’s interests, varies with time and it would have required more time to evolve itself in a more consistent and more suitable way, to converge toward the actual student model.

5.3.2 Adaptation Decision Making

In this section, we evaluate some different aspects of the Adaptation Decision Making mechanism. The question is:

RQ3: Are the adaptation decisions valid and meaningful, for the given state of the student model?

The adaptive mechanism is based on the building of a new LOS on the basis of the student model and on the parameters illustrated in Section 3; every time a student leaves a learning node after having taken a post-LN questionnaire, in order to measure her knowledge about the knowledge item associated to that particular learning node. 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. Fig. 8 shows all the students’ choices. The most important result is that 60 percent of students followed 100 percent of the suggested LOS while 6.67 percent followed more than 85 percent of the suggested LOS. More than 85 percent followed more than 60 percent of the suggested LOS.

In [30] and [31], we presented an evaluation of the quality of the new LOS proposed by the system on the basis of the student model and of the other parameters needed by the adaptive algorithm to run. In [30], we presented a first evaluation based on two LOS suggested 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 student models. The experimental results showed that 7.1 percent disagree, 7.1 percent neither agree nor disagree, 71.4 percent agree, and 14.4 percent strongly agree with the first didactic plan; 78.6 percent agree and 21.4 percent strongly agree with the second one. In the second paper [31], we addressed this problem alone by presenting an experimentation through six case studies.

![Fig. 7. The self-assessment frequency distribution. A 7-point Likert scale was used to have a more granular judgment.](image)

![Fig. 8. Analysis of the suggested LOS followed by students.](image)
where six new LOS were proposed on the basis of the student models in the software programming domain. A sample of 30 teachers were asked to assess the didactic validity of two proposed plans. More than 70 percent of the teachers gave a positive opinion of the proposed LOS. The domain of this evaluation was the programming domain. This evaluation can be considered valid in this context as well, because of the independence of the adaptive decision-making mechanism from the particular learning domain.

5.3.3 Discussion
In the Layered Evaluation, we analyzed the student modeling component and the adaptation decision making separately. Both for the student model and LOS validation, we obtained encouraging results: students showed to agree with their virtual student model proposed by the system, while in [30], [31], we showed that most teachers assessed the generated LOS positively.

5.4 User Attitudes and Affect Analysis
Here, we report some experimental data, gathered directly from students by means of a questionnaire on user attitudes and affect, submitted at the end of the course. This questionnaire consisted of 13 questions on the student’s satisfaction degree in the use of the LECOMPS system. In Fig. 9, we asked for the enjoyment degree in the use of the system. Both adaptive and nonadaptive students enjoyed the system. Fig. 10 shows the assessment of the graphical environment.

5.5 Navigation Analysis
We gathered some useful information concerning the interaction with the system during students’ navigation, as shown in Table 2.

<table>
<thead>
<tr>
<th>Modality</th>
<th>Visited Nodes</th>
<th>Navigation Time (min.)</th>
<th>Average Time per Node (min.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non Adaptive</td>
<td>15</td>
<td>39</td>
<td>3</td>
</tr>
<tr>
<td>Adaptive</td>
<td>13</td>
<td>38</td>
<td>3</td>
</tr>
</tbody>
</table>

In particular, we can see that the two learning modalities are almost identical in the number of visited nodes, time spent per node, and global navigation time. This is not in contradiction with the previous results, because, thanks to the WMW test, we can say that the quality of learning in the two modalities was somehow different, while by means of the U-Test and power analysis, we verified both the non-difference between the groups X and Y and a good power value.

5.6 Learning Styles Evolution
Finally, in this section, we discuss a relevant case study of a learner’s learning styles evolution in time, as expressed by the UPDATESM function of Fig. 2 shown in Section 3.1.2, to better emphasize the system behavior by means of its adaptive engine. Table 3 and Fig. 11 show a student’s learning styles evolution, taken from the log files for the StudentId = “1100”. In the first column of Table 3, all the learning nodes sequentially visited by the student are shown. The other four columns report all the changes in the Student’s learning style dimensions, starting from the first value, as measured by the Felder and Soloman test, i.e., $LS_{0,w-1100} = [-1, -5, -3, -3]$ and changing after each learning node activity. It is interesting to note how the four learning style dimensions change as the student performs her task according to the UPDATESM function on each node. In fact, this is a case study where the system heavily changes the student’s starting learning styles: from the starting $LS_0 = [-1, -5, -3, -3]$ to $LS_{10} = [2.49, -1.63, -0.34, 0.37]$. In particular, this student visited the same node “Development of Neorealism” three times, with its own $LS = [4.3, -2.4]$. In fact, the first two times the student fails the assessment and we can see that the learning styles move in the opposite direction of the learning styles of the node. When, the third time the student passes the test, the system modifies the student’s learning styles toward the learning styles of the node. Then, in Fig. 11, the student’s learning styles evolution is shown in a graphical form where is simpler to note how learning styles vary with time. For other interesting case studies, the reader can refer to [31].

6 CONCLUSIONS AND FUTURE WORK
We reported on an extended evaluation of LS-PLAN, a system devised to support the adaptive sequencing of learning material in a personalized course. The integration of LS-PLAN in the hypermedia module of LECOMPS allowed to manage a whole educational hypermedia system on Italian Neorealist Cinema.

The main contribution of our work is twofold: it deals with the methodological aspect and with the experimental one. From a methodological point of view, we followed two main guidelines: the classic experimental As a Whole plan, where two groups of users navigate With and Without the LS-PLAN support, and the Layered Evaluation plan, well
suited to measure both SM and Adaptive Decision Making soundness separately. We also presented three further evaluations: the students’ satisfaction rating, some navigation parameters as logged by the system, and a quick look into the adaptive algorithms behavior by means of a significant case study.

In the As a Whole experiment, we used nonparametric statistics, showing that users who navigate in the With modality show a knowledge that is 27.54 percent higher than the knowledge of those navigating in the Without modality. This first result strengthened our first research hypothesis.

Through the Layered Evaluation we acknowledged good results on the appropriateness of the SM and Adaptive Decision Making, on which the research questions RQ_2 and RQ_3 were positively confirmed.

The main lesson learned is that the evaluation of an adaptive system requires a complete experimental setup where all the main aspects should be taken into account.

In conclusion, the use of two measuring approaches, the Layered Evaluation and the As a Whole, was successful: through the former experimentation, we succeeded in calculating the added value of the adaptive component, which is very difficult indeed to compute through the latter; through the Layered Evaluation, we confirmed that the added value of the system was due to the adaptive engine.

As regards future work we plan to embed LS-PLAN into a state-of-the-art Learning Management System to put on trial our system in a more widely used system, exploring the possibility to fill in the adaptive capabilities gap currently present in e-learning platforms.

### References
