Accommodating Learning Styles in an Adaptive Educational System

Elvira Popescu, Costin Badica and Lucian Moraret
University of Craiova, A.I.Cuza 13, 200585 Craiova, Romania
E-mail: popescu_elvira@software.ucv.ro, badica_costin@software.ucv.ro

Keywords: intelligent e-learning application, adaptive educational system, learning style, learner modeling

Received: May 11, 2010

Integrating learning styles in adaptive educational systems is a relatively recent trend in technology enhanced learning. The rationale is that adapting courses to the learning preferences of the students has a positive effect on the learning process, leading to an increased efficiency, effectiveness and/or learner satisfaction. The purpose of this paper is twofold: i) to provide an extensive review of existing learning style-based adaptive educational systems (LSAES); ii) to propose an innovative system (called WELSA), which alleviates some of the encountered limitations. Specifically, WELSA is based on: i) a comprehensive set of learning style preferences; ii) an implicit and dynamic learner modeling method; iii) a dynamic adaptation approach. The system's architecture is presented, together with the main components responsible for its functionalities: authoring tool, data analysis tool and adaptation component. Encouraging experimental data are also reported.

Povzetek: V prispevku je podan pregled sistemov za učenje, ki se prilagajajo učencu, in nov sistem WELSA.

1 Introduction

An important class of intelligent applications in e-learning are the adaptive ones, namely those that aim at individualizing the learning experience to the real needs of each student. The rationale behind them is that accommodating the individual differences of the learners (in terms of knowledge level, goals, learning style, cognitive abilities, etc.) is beneficial for the student, leading to an increased learning performance and/or learner satisfaction. A common feature of these systems is that they build a model of learner characteristics and use that model throughout the interaction with the learner [3]. An adaptive system must be capable of managing learning paths adapted to each user, monitoring user activities, interpreting them using specific models, inferring user needs and preferences and exploiting user and domain knowledge to dynamically facilitate the learning process [4].

The idea dates back to 1995-1996, when the first intelligent and adaptive Web-based educational systems (AI-WBES) were developed [3]. Since then, both the intelligent techniques employed evolved and the range of learner characteristics that the systems adapt to expanded. A relatively recent characteristic that has started to be taken into account is the learning style of the student, i.e., the individual manner in which a person approaches a learning task, the learning strategies activated in order to fulfill that task. More formally, learning styles represent a combination of cognitive, affective and other psychological characteristics that serve as relatively stable indicators of the way a learner perceives, interacts with and responds to the learning environment [16].

For example, some learners prefer graphical representations and remember best what they see, others prefer audio materials and remember best what they hear, while others prefer text and remember best what they read. There are students who like to be presented first with the definitions followed by examples, while others prefer abstract concepts to be first illustrated by a concrete, practical example. Similarly, some students learn easier when confronted with hands-on experiences, while others prefer traditional lectures and need time to think things through. Some students prefer to work in groups, others learn better alone. These are just a few examples of the many different preferences related to perception modality, processing and organizing information, reasoning, social aspects, etc., all of which can be included in the learning style concept [24].

This paper deals with an intelligent learning environment that adapts to the learning style of the students, as its name suggests: WELSA - ***W***eb-based ***E***ducational system with ***L***earning ***S***tyle ***A***daptation. We start, in section 2, with an extensive review of related works, overviewing the adaptation techniques, as well as the modeling methods employed. Next, we introduce our innovative system, WELSA, based on: i) a comprehensive set of learning style preferences; ii) an implicit and dynamic learner modeling method; iii) a dynamic adaptation approach. The system architecture is presented in section 3, as well as an example of the platform at work. The following 3 sections present in more detail the main components responsible for the system's functionality: authoring tool (section 4), modeling component (section 5) and adaptation component (section 6). Finally, some conclusions are drawn in section 7.

2 Related works

In what follows, we will provide a summary of the state-of-the-art LSAES, classified from the point of view of the ***adaptation methods*** offered by these systems. Some of them combine adaptation provisioning based on several criteria: learning styles, knowledge level, goals, etc.; however, in what follows, we are only interested in the adaptation techniques used for learning style personalization. One of the most widely used techniques is the so-called ***fragment sorting*** [2], i.e., presenting the educational resources in an order considered most suitable for each student. So, basically, all the students are presented with the same learning resources, just ordered differently. This approach is used in several works, such as:

- [5] $\rightarrow$ The adaptation criteria in the CS383 system are represented by 3 constructs of the Felder-Silverman model (FSLSM) [9]: Sensing/Intuitive, Visual/Verbal, Sequential/Global. For each category of resources (i.e., hypertext, audio files, graphic files, digital movies, instructor slideshows, lesson objectives, note-taking guides, quizzes, etc.), the teacher has to mention its suitability (support) for each learning style (by rating it on a scale from 0 to 100). When a student logs into the course, a CGI executable loads the student profile (i.e., his/her learning style as resulted from answering a dedicated questionnaire); it then computes a unique ranking of each category of resources, by combining the information in the student's profile with the resource ratings. Next, the CGI dynamically creates an HTML page containing an ordered list of the educational resources, from the most to the least effective from the student's learning style point of view.

- [19] $\rightarrow$ The adaptation criteria in the INSPIRE system include the 4 learning styles in Honey and Mumford model [13]: Activist, Pragmatist, Reflector and Theorist. All learners are presented with the same knowledge modules, but their order and appearance (either embedded in the page or presented as links) differ for each learning style. Thus for Activists (who are motivated by experimentation and challenging tasks), the module "Activity" appears at the top of the page, followed by links to examples, theory and exercises. In case of Pragmatists (who are motivated by trying out theories and techniques), the module "Exercise" appears at the top of the page, followed by links to examples, theory and activities. Similarly, in case of Reflectors the order of modules is: examples, theory, exercises, and activities, while in case of Theorists the order is: theory, examples, exercises and activities. The system offers also the students the possibility to choose their preferred order of studying.

- [12] $\rightarrow$ The adaptation criteria are represented by three FSLSM dimensions (Active/Reflective, Sensing/Intuitive, Sequential/Global). The authors propose an add-on for Moodle Learning Management System [18], which supplies the required adaptation. More specifically, it provides an individualized sequence and number of learning objects of each type (i.e., examples, exercises, self assessment tests, content objects).

Another adaptation technique is to ***customize the system's interface*** according to students' preferences. This technique is used for example in [6]. The adaptation criterion is represented by the Felder-Silverman learning style model. The interface is adaptively customized: it contains 3 pairs of widget placeholders (text/image, audio/video, Q&A board/Bulletin Board), each pair consisting of a primary and a secondary information area. The space allocated on the screen for each widget varies according to the student's FSLSM learning style: e.g., for a Visual learner the image data widget is located in the primary information area, which is larger than the text data widget; the two widgets are swapped in case of a Verbal learner. Similarly, the Q&A Board and Bulletin Board are swapped in case of the Active versus Reflective learners.

A similar approach is used by [1]. However, besides layout customization, they also alter the sequencing and structure of the learning content, as well as the navigation options. The adaptation criterion is represented by the FSLSM Sequential / Global preference. The pages for Global students contain diagrams, table of contents, overview of information, summary, while pages for Sequential learners only include small pieces of information, and Forward and Back buttons.

A more complex adaptation approach is employed by [30]. They use both adaptive presentation technique and adaptive navigation support to individualize the information and the learning path to the field dependence (FD)/field independence (FI) characteristic of the students [32]. Specifically, the AES-CS system uses ***conditional text*** and ***page variants*** to present the information in a different style: from specific to general in case of FI learners (who have an analytic preference) and from general to specific in case of FD learners (who have a global preference). AES-CS offers also two control options: program control for FD learners, by means of which the system guides the learner through the learning material; learner control for FI learners, by means of which the learners can choose their own learning paths, through a menu. Since FD learners benefit more from instructions and feedback, an additional frame at the bottom of the page is used to provide them with explicit directions and guidance. This frame is missing in case of FI learners, who prefer few instructions and feedback. Similarly, in case of self-assessment tests, the feedback provided for FI learners is less extensive than in case of FD learners. Finally, FD learners are offered two navigational tools in order to help them structure the learning material and create the big picture: a concept map (a visual representation of the domain concepts and the relations between them) and a graphic path indicator (presenting the current, the previous and the next topic). Furthermore, AES-CS allows students to modify the adaptation options provided by the system, making their own choices between program / learner control, minimal / maximal feedback, etc.

Another approach is the ***adaptive selection of learning objects***, among the set of equivalent ones (from the point of view of the domain concept that they explain). The learning object (LO) that best suits the learning style of the current student is included in the learning path. Two papers that use this method are:

- [27] $\rightarrow$ The adaptation criteria include the four FSLSM dimensions. Each LO is manually annotated by the teacher using IMS Metadata Standard [14]. Each of the possible "Learning Resource Type" metadata values (i.e., "Exercise", "Simulation", "Questionnaire", "Diagram", "Figure", "Graph", "Index", "Slide", "Table", "Narrative Text", "Exam", "Experiment", "ProblemStatement", "SelfAssesment") are classified with the help of pedagogic experts according to the Felder and Silverman's teaching styles. First, the system finds the set of necessary domain concepts to be taught to the current student, based on the domain ontology and student's knowledge level. Next, for each domain concept, the set of LOs that explain it are found; the system selects one of these LOs taking into account the value of the attribute "Learning Resource Type" and trying to minimize the distance between the learning style and teaching style (interpreted as Euclidian distance).

- [17] $\rightarrow$ Again, the adaptation criterion is represented by the Felder-Silverman model. Each learning object is annotated by the teacher with a set of weights corresponding to its suitability for each of the 4 FSLSM dimensions. First, the system automatically generates a personalized learning path by means of a planner which takes into account the student's knowledge level and her FSLSM score. At each step, the system can output a new Learning Object Sequence, in case the student model has changed. For each knowledge item on the learning path, the system selects the associated LO which is the most suited for the learning style of the student, based on the assigned weights (i.e., having the smallest Euclidian distance from the student's learning style).

A more ***generic adaptation approach*** is proposed by Stash [28]. She uses an XML Learning Style Adaptation Language, called LAG-XSL, based on the LAG language (i.e., generalized adaptation model for generic adaptive hypermedia authoring [8]). LAG-XSL is a high level language, including adaptation actions such as: selection of different representations of concepts (media, level of difficulty, type of activity) and sorting of concepts. By means of these actions, authors can define their own adaptation strategies for their own learning styles. However, there is a limitation in the types of strategies that can be defined and consequently in the set of learning preferences that can be used. Paper [28] includes examples of 3 such instructional strategies, for Verbalizer versus Imager style, Global versus Analytic style and Activist versus Reflector style.

As far as the ***method for identifying the learning style*** of the student is concerned, the existing LSAES can be classified in two categories:

1. those that use an explicit modeling method (i.e., rely on the measuring instruments associated to the learning style models for diagnosing purposes)
2. those that use an implicit modeling method (i.e., based on the analysis of students' observable behavior).

The main advantages of the second category of systems are:

1. they don't require any additional work form the part of the students (for filling in the questionnaires)
2. they overcome the psychometric flaws of the traditional measuring instruments (which sometimes lack internal consistency, test-retest reliability or construct and predictive validity)
3. the student model can be continuously updated - it doesn't have to be static, created at the beginning of the course and stored once and for all.

Examples of works that fall in the first category are: [1], [5], [17], [19], [30], [31]. Examples from the second category include: [7], [10], [11], [12], [20], [27], [28], [29], [33].

In this paper we report a system (WELSA), which uses an implicit modeling method, combined with adaptive sorting and adaptive annotations techniques. Furthermore, WELSA is based not on a single learning style model (as all the systems included above), but on a complex of features extracted from several such learning style models. Finally, WELSA was thoroughly tested and experimental data is available regarding the accuracy of the modeling method as well as the efficiency and effectiveness of the adaptation on the learning process.

3 WELSA Overview

WELSA's functionalities are primarily addressed at the students, who can learn by browsing through the course and performing the instructional activities suggested (play simulations, solve exercises, etc.). They can also communicate and collaborate with their peers by means of the forum and chat. Students' actions are logged and analyzed by the system, in order to create accurate learner models. Based on the identified learning preferences and the built-in adaptation rules, the system offers students individualized courses. WELSA provides also functionalities for the teachers, who can create courses by means of the dedicated authoring tool; they can also set certain parameters of the modeling process, so that it fits the particularities of their course.

Figure 1 shows how WELSA appears for a learner who is studying a course on Artificial Intelligence (more specifically the chapter on "Constraint Satisfaction Problems", based on the classical textbook of Poole, Mackworth and Goebel [21]).

A few notes should be made regarding the course pages: the first resource (LO) on the page is entirely visible (expanded form), while for the rest of LOs only the title is shown (collapsed form). Of course, the student may choose to expand or collapse any resource, as well as lock them in an expanded state by clicking the corresponding icons. Also, there are specific icons associated to each LO, depending on its instructional role and its media type, in order to help the learner browse more effectively through the resources. Finally, navigation can be done by means of the Next and Previous buttons, the course outline or the left panel with the chapter list.

3.1 Architecture

The overall architecture of WELSA is illustrated in Fig. 2. WELSA is composed of three main modules:

- an authoring tool for the teachers, allowing them to create courses conforming to the internal WELSA format (XML-based representation)
- a data analysis tool, which is responsible for interpreting the behavior of the students and consequently building and updating the learner model, as well as providing various aggregated information about the learners
- a course player (basic learning management system) for the students, enhanced with two special capabilities: i) learner tracking functionality (monitoring the student interaction with the system); ii) adaptation functionality (incorporating adaptation logic and offering individualized course pages).

The three modules will be presented in more details in the next three sections.

As far as the implementation is concerned, Java-based and XML technologies are employed for all WELSA components. Apache Tomcat 6.0 is used as HTTP web server and servlet container and MySQL 5.0 is used as DBMS.

4 WELSA authoring tool

The course structure that we propose in WELSA is a hierarchical one: each course consists of several chapters, and each chapter can contain several sections and subsections. The lowest level subsection contains the actual educational resources. Each such elementary learning object corresponds to a physical file and has a metadata file associated to it [22]. These metadata are independent of any learning style; they describe the LO from the point of view of media type, format, instructional role, abstractness level, prerequisite, hierarchical and similarity relations with other LOs. Apart from being widely used for organizing the teaching materials, this approach also insures a high reusability degree of the educational resources. Furthermore, due to the fine granularity level of the LOs, a fine granularity of adaptation actions can also be envisaged. Finally, since each LO has a comprehensive metadata file associated to it, we know all the information about the learning resource that is accessed by the learner at a particular moment, so we can perform a detailed learner tracking.

In order to support the teacher in creating courses conforming to WELSA internal format, we have designed a course editor tool, which allows authors to easily assemble and annotate learning resources, automatically generating the appropriate file structure. It should be noted that WELSA course editor does not deal with the creation of actual content (text, images, simulations, etc.) - a variety of existing dedicated tools can be used for this purpose (text editors, graphics editors, HTML editors, etc.). Instead, WELSA course editor provides a tool for adding metadata to existing learning resources and defining the course structure (specifying the order of resources, assembling learning objects in pages, sections and subsections). The teacher can define this chapter structure in a simple and intuitive way, by using the course editor, as shown in Fig. 3. The corresponding XML files are subsequently generated by the application and stored on the server [23].

5 WELSA analysis tool (modeling component)

The adoption of a suitable taxonomy of learning styles plays an important role in the overall quality of the system. The result of the adaptation process can only be as accurate and comprehensive as the underlying student model. As mentioned in section 2, WELSA is based not on a single learning style model, like the rest of the similar systems, but on a complex of features extracted from several such learning style models (called ULSM - Unified Learning Style Model). This model integrates characteristics related to: perception modality, way of processing and organizing information as well as motivational and social aspects (e.g., *Visual / Verbal, Abstract / Concrete, Serial / Holistic, Active experimentation / Reflective observation, Individual work / Team work, Intrinsic motivation / Extrinsic motivation*). A detailed description of the ULSM characteristics, together with the model's rationale and advantages, is included in [25].

For the identification of these ULSM preferences, WELSA uses an implicit modeling mechanism, by analyzing the interaction of the students with the educational system, in the form of behavioral patterns. Once the learner actions are recorded by the course player, they have to be processed by the Analysis tool, in order to yield the learning preferences of the students. The modeling mechanism

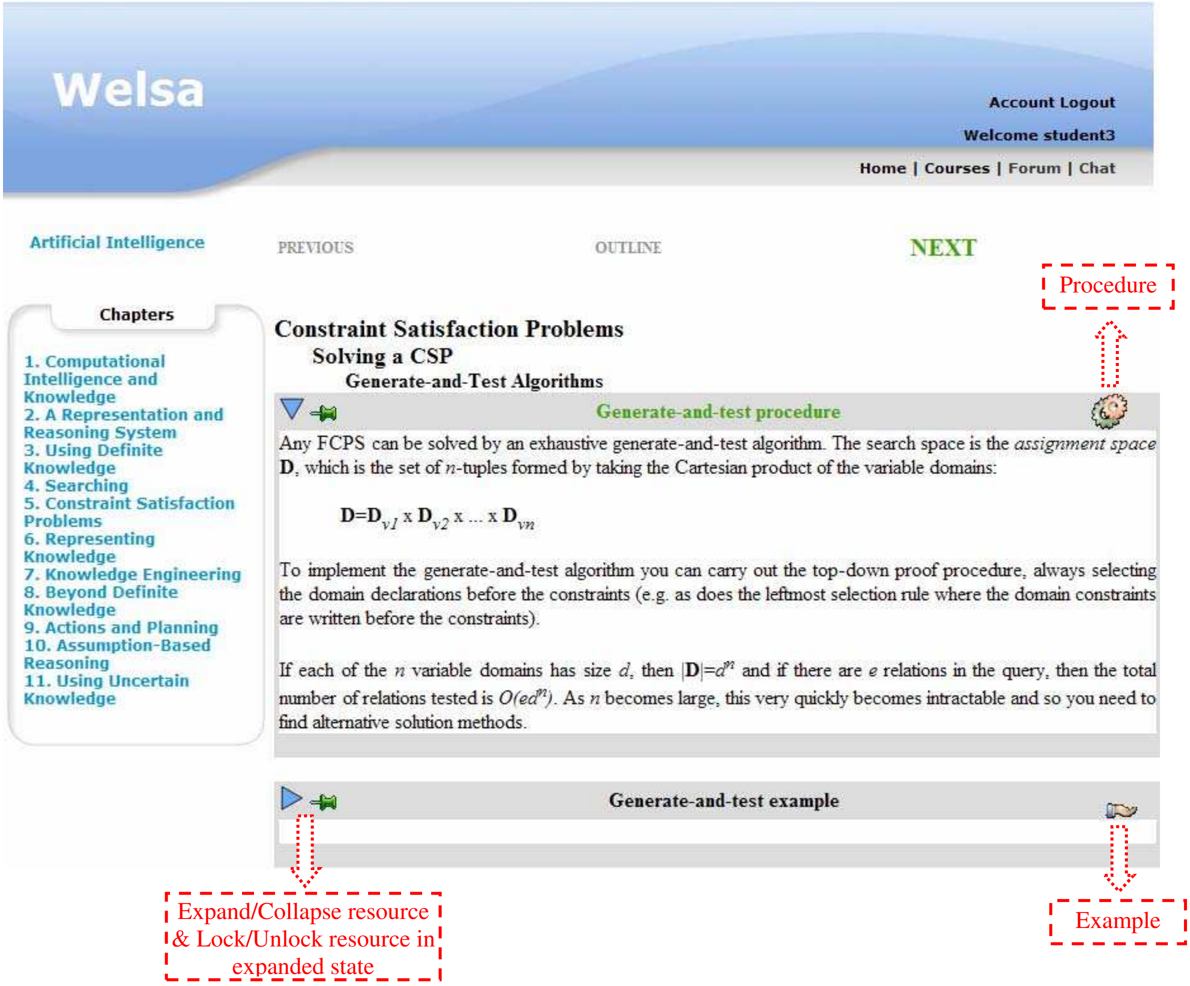

Figure 1: A snapshot of WELSA (student view)

is depicted in Fig. 4.

In order to compute the pattern values, a pre-processing phase of the raw data (i.e., the student actions and the associated timestamps) is necessary. The first step is to compute the duration of each action for each student, eliminating the erroneous values (for example, accessing the outline for more than 3 minutes means that the student actually did something else during this time). Next, the access time for each LO is computed, again filtering the spurious values (for example, an LO access time of less than 3 seconds was considered as random or a step on the way to another LO and therefore not taken into account). The data are then aggregated to obtain the pattern values for each student (e.g., total time spent on the course, total number of actions performed while logged in, time spent on each type of LO, number of hits on each category of LOs, the order of accessing the LOs, the number of navigation actions of a specific type, the number of messages in chat / forum, etc.). The reliability levels of these patterns are calculated as well (i.e., the larger the number of available relevant actions, the more reliable the resulted pattern). Next, the Analysis tool computes the ULSM preferences values, using modeling rules based on the pattern values, their reliability levels and their weights, as detailed in [24]. It should be noted that these rules also take into account the specificities of each course: the pattern thresholds as well as the importance of each pattern may vary with the structure and subject of the course. Therefore, the teachers should have the possibility to adjust the predefined values to correspond to the particularities of her/his course or even to eliminate some of the patterns, which are not relevant for that course. This is why the Analysis tool has a configuration option, which allows the teacher to modify the weight and threshold values, as seen in Fig. 5.

Beside the function of diagnosing the student learning preferences and correspondingly updating the learner model, the Analysis tool also offers various aggregated data that can be used for comparisons and statistical purposes. These tasks are accomplished by a researcher who interacts with the Analysis tool in the experimental version of WELSA. All the intermediate data (duration of learner actions, pattern values, pattern thresholds, reliability and confidence values) can be visualized by the researcher. Furthermore, at researcher's request, the analysis tool com-

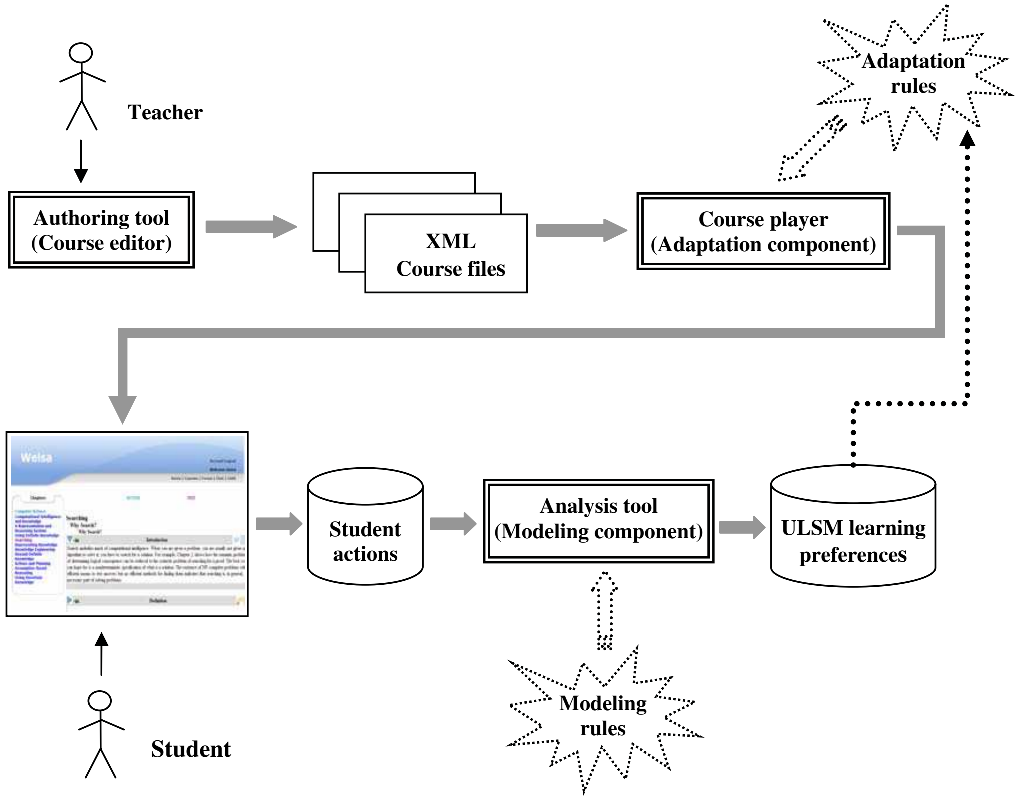

Figure 2: Overall WELSA architecture

putes and displays aggregated information, such as the total number of students with each ULSM preference, the total and average number of student actions, the average reliability and confidence values, etc. These data can be used for further analysis (e.g., by processing them in a dedicated statistical package). The roles and interactions of the actors with the Analysis tool are illustrated in Fig. 6.

In order to test the modeling method implemented in the Analysis tool, an experiment involving 71 undergraduate students was realized. The learners studied an AI course module on "Search strategies and solving problems by search" and all of their interactions with WELSA were recorded by the course player. Next, the Analysis tool computed the values of the behavioral patterns and applied the modeling rules, inferring the ULSM learning preferences of each student. In order to evaluate the validity of our modeling method, the results obtained by the Analysis tool (implicit modeling method) were compared with the reference results obtained using the ULSM questionnaire (explicit modeling method). Good precision results were obtained, with an average accuracy of 75.70%, as reported in [24].

6 WELSA course player (adaptation component)

WELSA course player is responsible with the generation of individualized web pages for each student; furthermore, it incorporates some basic LMS (learning management system) functions, such as: administrative support (registration and authentication) and communication and collaboration tools (discussion forum, chat).

Another function of the course player is to track student actions (down to click level) and record them in a database for further processing by the Analysis tool. This is done with the help of JavaScript code added to the HTML page, coupled with Ajax technology. Thus the application can communicate with the web server asynchronously in the background, without interfering with the display and behavior of the existing page.

In what follows we will give some details regarding the most important functionality of the course player, namely the adaptation mechanism, which allows the dynamic generation of individualized courses for each student.

Once the students' learning preferences are identified by the Analysis tool, the next step is to associate adaptation actions that are best suited for each preference. The development of these adaptation rules was a delicate task, since it involved interpretation of the literature in order to identify the prescriptive instructional guidelines. Indeed, apart

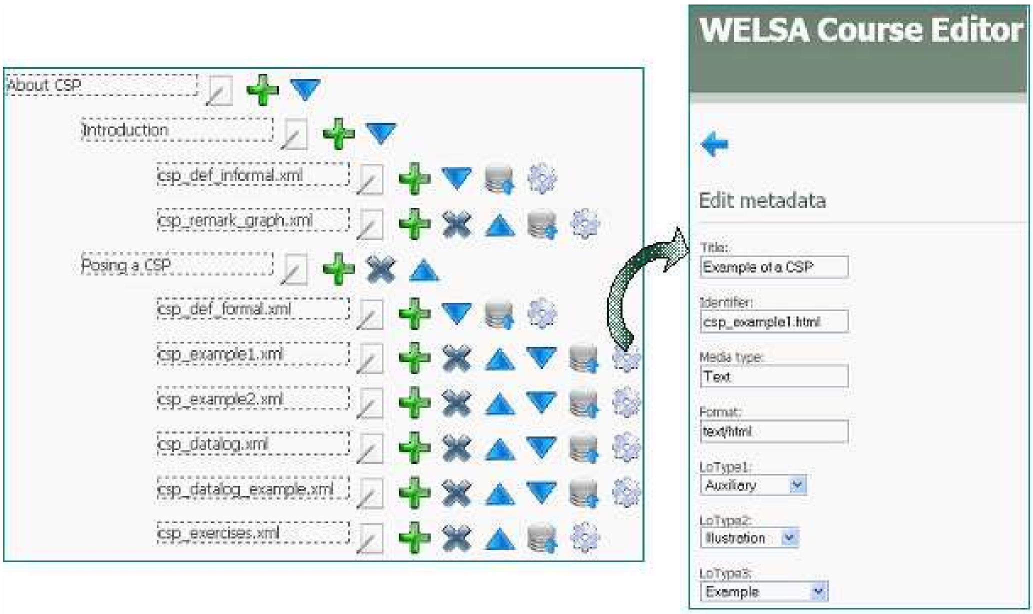

Figure 3: Snapshot of WELSA authoring tool: editing course structure (left-hand side) & editing metadata (right-hand side)

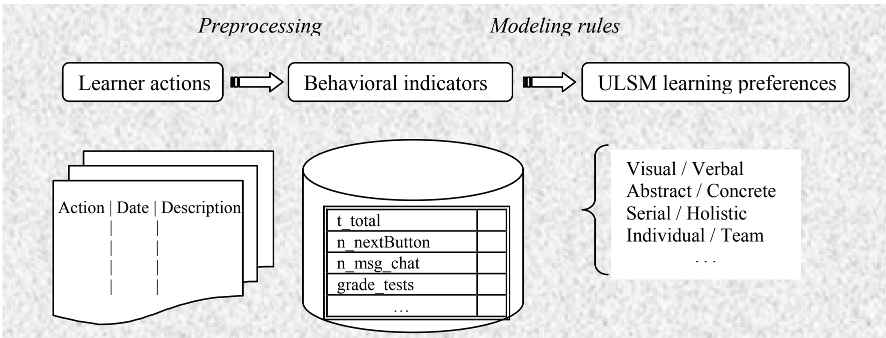

Figure 4: WELSA learner modeling mechanism

from defining the characteristics of the learners belonging to each learning style, for most of the models there are proposed teaching practices that effectively address the educational needs of students with the identified styles. However, as noted in [15], "learning styles models are usually rather descriptive in nature, in the sense that they offer guidelines as to what methods to use to best attain a given goal; they are not usually prescriptive in the sense of spelling out in great detail exactly what must be done and allowing no variation". Starting from these teaching methods (which only include a traditional learning view), enhancing them with e-learning specific aspects (technology-related preferences) and inspiring from other works that dealt with learning style based adaptation (as mentioned in section 2), we extracted the adaptation rules for our LSAES.

More specifically, we decided to use adaptive sorting and adaptive annotation techniques. The LOs are placed in the page in the order which is most appropriate to each learner; additionally, a "traffic light metaphor" was used to differentiate between recommended learning objects (LOs) (with a highlighted green title), standard LOs (with a black title) and not recommended LOs (with a dimmed light grey title) [26]. It should be mentioned however that the learning path suggested by the system is not compulsory: it is simply a recommendation that the student may choose to follow or not. We consider that offering control to students, instead of strictly guiding them, is a more flexible and rewarding pedagogical approach.

The adaptation mechanism is illustrated in Fig. 7, with a fragment of a Web page from an AI course generated for a student with a preference towards *Concrete, practical examples* rather than *Abstract concepts and generalizations*. The page is dynamically composed by selecting the appropriate LOs (mainly of type Example), each with its own status (highlighted in case of LOs of type Example and standard in case of LOs of type Definition) and ordered

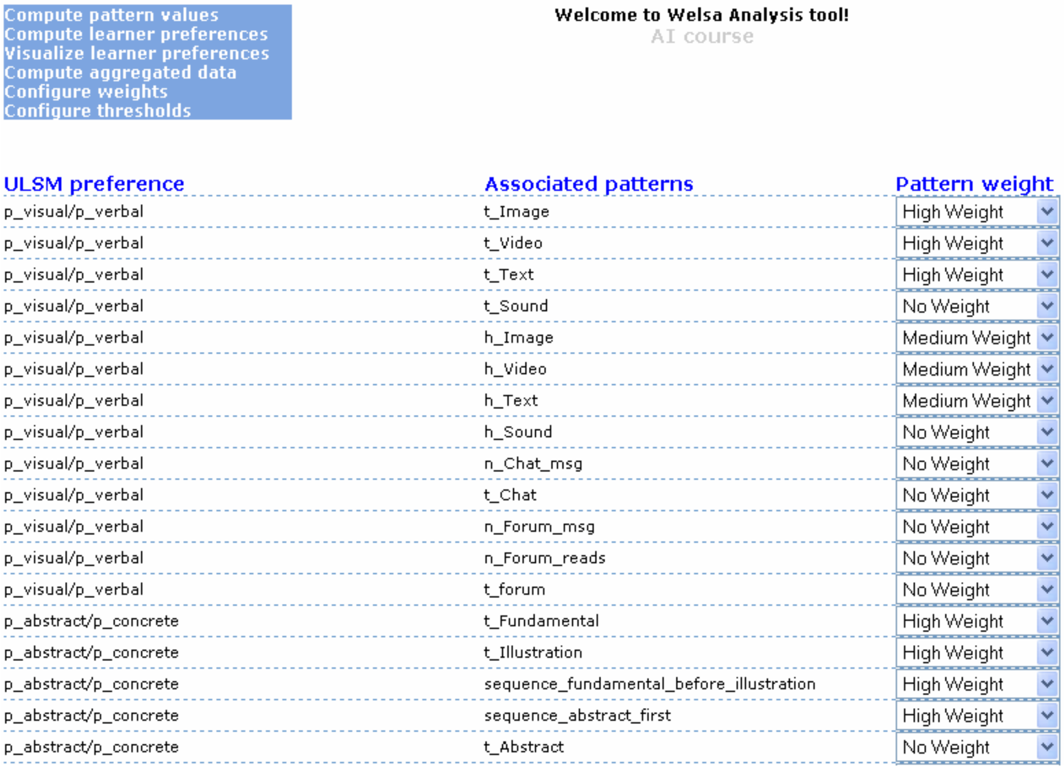

Figure 5: A snapshot from WELSA Analysis tool, illustrating the configuration options

correspondingly (first the notion of "Constraint satisfaction problem" is illustrated by means of two examples and only then a definition is provided).

Formally, the corresponding adaptation rules are included in Fig. 8. Note that *LoType* refers to the instructional role of the LO, as described in the metadata. More details regarding the LO indexing can be found in [22].

Adaptation rule for learners with "Abstract" preference
IF
$p\_abstract \in Pref(L)$
THEN
Sort $LoType$ {*Fundamental, Illustration*}
Highlight $LoType$ {*Fundamental*}

Adaptation rule for learners with "Concrete" preference
IF
$p\_concrete \in Pref(L)$
THEN
Sort $LoType$ {*Illustration, Fundamental*}
Highlight $LoType$ {*Illustration*}

Figure 8: Adaptation rules for *Abstract/Concrete* preference

The adaptation component consists of a Java servlet which automatically generates the individualized web page, each time an HTTP request is received by the server, as illustrated in Fig. 9. WELSA doesn't store the course web pages but instead generates them on the fly, following the structure indicated in the XML course and chapter files.

The adaptation servlet queries the learner model database, in order to find the ULSM preferences of the current student. Based on these preferences, the servlet applies the corresponding adaptation rules and generates the new HTML page. These adaptation rules involve the use of LO metadata, which as already stated in section 4, are independent of any learning style. However, they convey enough information to allow for the adaptation decision making (i.e., they include essential information related to the media type, the level of abstractness, the instructional role, etc.). Next the web page is composed from the selected and ordered LOs, each with its own status (highlighted, dimmed or standard).

This dynamic adaptation mechanism reduces the workload of authors, who only need to annotate their LOs with standard metadata and do not need to be pedagogical experts (neither for associating LOs with learning styles, nor for devising adaptation strategies). The only condition for LOs is to be as independent from each other as possible, without cross-references and transition phrases, to insure that the adaptation component can safely apply reordering techniques. Obviously, there are cases in which changing the order of the learning content is not desirable; in this case the resources should be presented in the predefined order only, independently of the student's preferences (the teacher has the possibility to specify these cases by means

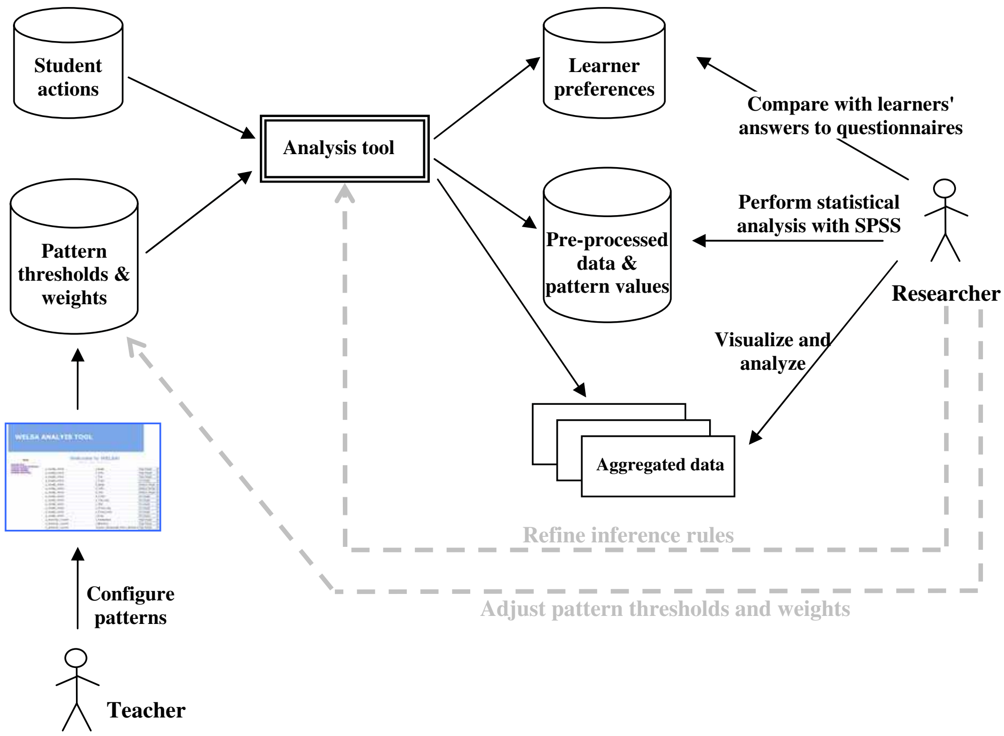

Figure 6: Users' interaction with the Analysis tool

of the prerequisites mechanism included in the metadata).

The validity and effectiveness of our adaptation approach were empirically confirmed by means of an experiment involving 64 undergraduate students in the field of Computer Science. The students were split in two groups: one which was provided with a matched version of the course (further referred to as "matched group") and one which was provided with a mismatched version of the course (further referred to as "mismatched group"), with respect to the students' learning preferences.

The objective evaluation consisted in performing a statistical analysis on the behavioral patterns exhibited by the students, comparing the values obtained for the matched and mismatched groups in order to find significant differences. The results showed that the matched adaptation approach increased the efficiency of the learning process, with a lower amount of time needed for studying and a lower number of randomly accessed educational resources (lower level of disorientation). The effectiveness of the matched adaptation and its suitability for addressing students' real needs are also reflected in the statistically significant higher time spent on recommended versus not recommended resources, as well the higher number of accesses of those recommended learning objects. Finally, the recommended navigation actions were followed to a larger extent than the not recommended ones.

As far as students' subjective evaluation of the system is concerned (as assessed by means of an opinion questionnaire), the students in the matched group reported significantly higher levels of enjoyment, overall satisfaction and motivation, compared to their mismatched peers. The overall results of the experimental study are very promising, proving the positive effect that our adaptation to learning styles has on the learning process. However, in order to allow for generalization, the system should be tested on a wider scale, with users of variable age, field of study, background knowledge and technical experience, which is one of our future research directions. Further details regarding the evaluation process can be found in [26].

7 Conclusion

The WELSA system described in this paper is an intelligent e-learning platform, aimed at adapting the course to the learning preferences of each student. We opened this paper with an extensive review of related LSAES. Starting from the existing systems, we introduced an innovative approach, based on an integrative set of learning preferences (ULSM). The technical and pedagogical principles behind WELSA were presented, focusing on the three main modules of the system. The learner modeling and adaptation methods were briefly introduced, together with their realization in WELSA.

As future work, improvements could be envisaged for each of the three main components. The authoring tool could be extended with an import/export facility, allowing for conversion between various course formats and standards (e.g., SCORM, IMS LD, etc.). The modeling component could also be extended to take into account the per-

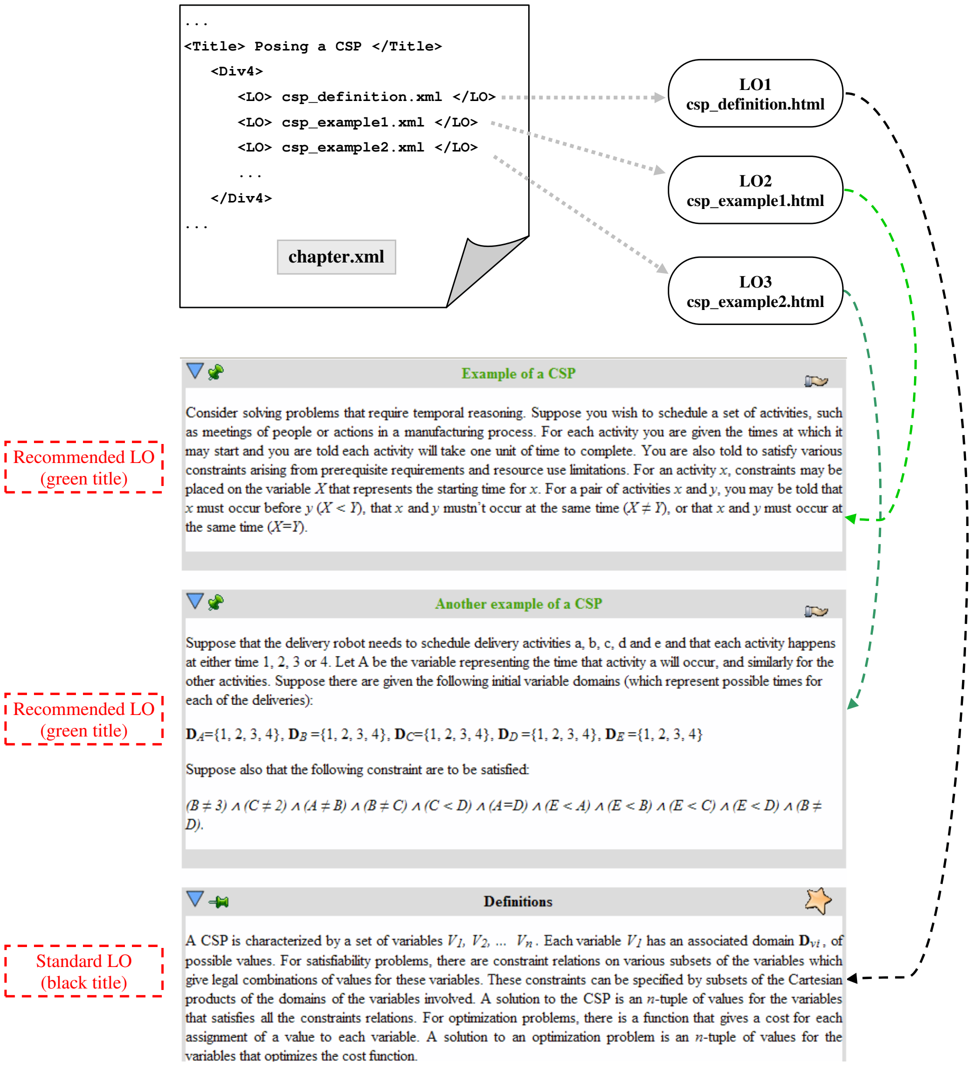

Figure 7: Composing a page from elementary LOs for a student with *Concrete* preference

turbations introduced by adaptation on students' actions; students' behavior in the adaptive version could be used as a valuable feedback on the effect of adaptation. Finally, the course player could incorporate a wider variety of adaptation actions, including also collaboration level adaptation techniques which are currently out of the scope of the system. In this respect, a wider range of communication and collaboration tools should be included in the system, including social software applications (e.g., blog, wiki, social bookmarking tool, etc.). Extending WELSA into a social and adaptive learning environment would be a challenging research direction.

8 Acknowledgment

This work was partially supported by the strategic grant POSDRU/89/1.5/S/61968, Project ID 61968 (2009), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007 – 2013.

References

[1] Bajraktarevic, N., Hall, W., Fullick, P. (2003) Incorporating learning styles in hypermedia environment: Empirical evaluation. *Proc. Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems*, pp. 41–52.

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

[3] Brusilovsky, P., Peylo, C. (2003) Adaptive and Intelligent Web-based Educational Systems. *International Journal of Artificial Intelligence in Education*, 13 (2-4), pp. 159–172.

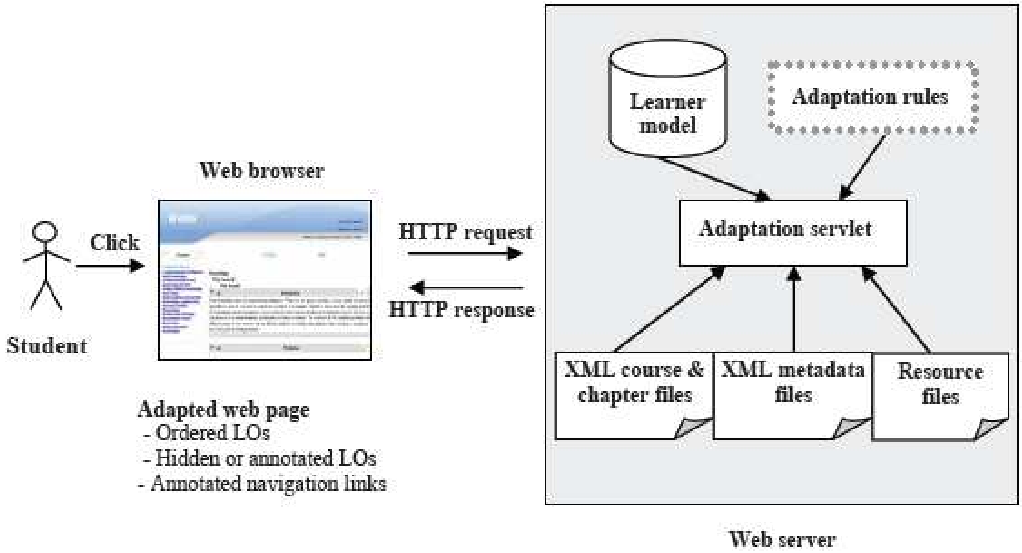

Figure 9: Adaptation component schematic architecture

[4] Boticario, J.G., Santos, O.C., van Rosmalen P. (2005) Issues in Developing Standard-based Adaptive Learning Management Systems. *EADTU 2005 Working Conference: Towards Lisbon 2010: Collaboration for Innovative Content in Lifelong Open and Flexible Learning*.

[5] Carver, C. A., Howard, R. A., Lane, W. D. (1999) Enhancing student learning through hypermedia courseware and incorporation of student learning styles. *IEEE Transactions on Education*, 42, pp. 33–38.

[6] Cha, H. J., Kim, Y. S., Lee, J. H., Yoon, T. B. (2006) An Adaptive Learning System with Learning Style Diagnosis based on Interface Behaviors. *Workshop Proceedings of Intl. Conf. E-learning and Games (Edutainment 2006)*.

[7] Cha, H. J., Kim, Y. S., Park, S. H., Yoon, T. B., Jung, Y. M., Lee J. H. (2006) Learning styles diagnosis based on user interface behaviors for the customization of learning interfaces in an intelligent tutoring system. *Procs. ITS 06*. Lecture Notes in Computer Science, Vol. 4053, Springer, pp. 513–524.

[8] Cristea, A., Calvi, L. (2003) The Three Layers of Adaptation Granularity. *Proc. UM 2003*, pp. 4–14.

[9] Felder, R. M., Silverman, L. K. (1988) Learning and Teaching Styles in Engineering Education. *Engineering Education*, 78(7). Preceded by a preface in 2002: `http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/LS-1988.pdf`.

[10] Garcia, P., Amandi, A., Schiaffino, S., Campo, M. (2007) Evaluating Bayesian NetworksŠ Precision for Detecting StudentsŠ Learning Styles. *Computers & Education*, 49(3), pp. 794–808.

[11] Gilbert, J.E., Han, C.Y. (1999) Adapting instruction in search of Śa significant differenceŠ. *Journal of Network and Computer Applications*, 22(3), pp. 149–160.

[12] Graf, S. (2007) *Adaptivity in Learning Management Systems Focussing on Learning Styles*. PhD Thesis, Vienna University of Technology, Austria.

[13] Honey, P., Mumford, A. (2000) *The Learning Styles Helper's Guide*. Maidenhead: Peter Honey Publications Ltd.

[14] IMS Metadata Standard (2010) `http://www.imsglobal.org/metadata/`.

[15] Karagiannidis, C., Sampson, D. (2004) Adaptation Rules Relating Learning Styles Research and Learning Objects Metadata. *Proc. Workshop on Individual Differences in Adaptive Hypermedia in AH2004*, pp. 60-69.

[16] Keefe, J.W. (1979) Learning style: an overview. *NASSP's Student Learning Styles: Diagnosing and Prescribing Programs*, pp. 1–17.

[17] Limongelli, C., Sciarrone, F., Temperini, M., Vaste, G. (2009) Adaptive Learning with the LS-Plan System: A Field Evaluation. *IEEE Transactions on Learning Technologies* 2(3), pp. 203–215.

[18] Moodle (2010) `http://moodle.org`.

[19] Papanikolaou, K.A., Grigoriadou, M., Kornilakis, H., Magoulas, G.D. (2003) Personalizing the interaction

in a Web-based educational hypermedia system: the case of INSPIRE. *User-Modeling and User-Adapted Interaction*, 13, pp. 213–267.

[20] Paredes, P., Rodriguez, P. (2004) A Mixed Approach to Modelling Learning Styles in Adaptive Educational Hypermedia. *Advanced Technology for Learning*, 1(4), pp. 210–215.

[21] Poole, D., Mackworth, A., Goebel, R. (1998) *Computational Intelligence: A Logical Approach*. Oxford University Press.

[22] Popescu, E., Badica, C., Trigano, P. (2008) Description and organization of instructional resources in an adaptive educational system focused on learning styles, *Procs. IDC 2007*, Studies in Computational Intelligence, Vol. 78, Springer, pp. 177–186.

[23] Popescu, E., Trigano, P., Badica, C., Butoi, B., Duica, M. (2008) A Course Authoring Tool for WELSA Adaptive Educational System. *Proc. ICCC 2008*, pp. 531–534.

[24] Popescu, E. (2009) Diagnosing Students' Learning Style in an Educational Hypermedia System. *Cognitive and Emotional Processes in Web-based Education: Integrating Human Factors and Personalization*, Advances in Web-Based Learning Book Series, IGI Global, pp. 187–208.

[25] Popescu, E. (2010) A Unified Learning Style Model for Technology-Enhanced Learning: What, Why and How?. *International Journal of Distance Education Technologies*, 8(3), IGI Global, pp. 65–81.

[26] Popescu, E. (2010) Adaptation Provisioning with respect to Learning Styles in a Web-Based Educational System: An Experimental Study. *Journal of Computer Assisted Learning*, 26(4), Wiley, pp. 243–257.

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

[28] Stash, N. (2007) *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*. PhD Thesis, Eindhoven University of Technology, Netherlands.

[29] Stathacopoulou, R., Grigoriadou, M., Samarakou, M., Mitropoulos, D. (2007) Monitoring studentsŠ actions and using teachersŠ expertise in implementing and evaluating the neural network-based fuzzy diagnostic model. *Expert Systems with Applications*, 32, pp. 955–975.

[30] Triantafillou, E., Pomportsis, A., Demetriadis, S. (2003) The design and the formative evaluation of an adaptive educational system based on cognitive styles. *Computers & Education*, 41, pp. 87–103.

[31] Wang, T., Wang, K., Huang, Y. (2008) Using a Style-based Ant Colony System for Adaptive Learning. *Expert Systems with Applications*, 34(4), pp. 2449–2464.

[32] Witkin, H.A. (1962) *Psychological Differentiation: Studies of Development*. New York: Wiley.

[33] Wolf, C. (2002) iWeaver: Towards an Interactive Web-Based Adaptive Learning Environment to Address Individual Learning Styles. *European Journal of Open, Distance and E-Learning*. Available at: `http://www.eurodl.org/materials/contrib/2002/2HTML/iWeaver.htm`.