Architecture Trends of Adaptive Educational Hypermedia Systems: The Case of the MATHEMA

Alexandros Papadimitriou

Department of Electrical and Electronic Engineering, School of Pedagogical and Technological Education, Athens, Greece

Email address:

apapadim@di.uoa.gr

Abstract: The aim of this article is to present the general architecture trends of Web-based Adaptive Educational Hypermedia Systems (AEHSs) and to give a complete description of architecture of the AEHS MATHEMA. In the beginning, a related work on the architecture trends of Web-based AEHSs is presented. Then, a description of the aspects of the MATHEMA is done regarding both its pedagogical and technological part. Next, one-on-one unit is presented separately and their functions are generally described with respect to the adaptive Web technologies used. Research has shown that students of high schools increase their performance by studying through the AEHS MATHEMA. Also, the formative evaluation of AEHS MATHEMA by students of the Department of Informatics and Telecommunications of the University of Athens, Greece, has shown, with the exception of other things, that all its functions are useful and easy to use.

Keywords: Adaptive Educational Hypermedia Systems, Architecture Trends of AEHSs, Adaptive Web Technologies

1. Introduction

Adaptive Hypermedia Education Systems (AEHSs) can be considered as the answer to the problems of traditional hypermedia systems. They build a personalized model of the learner and apply this model to adapt the content and/or the appearance of the hypermedia, according to the learner's specific characteristics, such as the cognitive goal, level of knowledge, pre-existing or prior knowledge, interests, preferences, stereotypes, cognitive preferences and cognitive or learning style [4].

In the Web-based AEHSs, several *adaptive and intelligent techniques* have been applied to introduce adaptation, such as [4]:

(a) *Curriculum Sequencing*: It helps the learner to follow an optimal path through the learning material.

(b) *Adaptive Presentation*: It adapts the content presented in each hypermedia node according to specific characteristics of the learner.

(c) *Adaptive Navigation Support*: It adapts the link structure in such a way that the learner is guided towards interesting and relevant information, kept away from non-relevant information either by suggesting the most relevant links to follow or by providing adaptive comments to visible links.

(d) *Meta-adaptive Navigation Support*: It selects or suggests the most appropriate adaptive navigation technique that suits the given learner best relatively to the given context, either by observing and evaluating the success of each technique in different contexts and the resulting learning from these observations, or by assisting the learner in selecting the navigation technique that best suits to him or her.

(e) *Interactive Problem Solving Support*: It provides the learner with intelligent help on every step of problem solving, by giving a hint to executing the next step for the learner.

(f) *Intelligent Analysis of Learner's Solutions*: It uses intelligent analysers that not only tell the learner whether the solution is correct but also it tells him/her what exactly is wrong or incomplete.

(g) *Example-based Problem Solving Support*: It helps the learners in solving new problems, not by articulating their errors, but by suggesting them relevant successful problem solving cases, chosen from their earlier experience.

(h) *Adaptive Collaboration Support; Adaptive Group Formation and/or Peer Help*: These techniques support the collaboration process either just like the interactive problem solving support systems assist an individual learner in

solving a problem, or they use knowledge about possible collaborating peers in order to form a matching group relatively to the kind of the collaborative task.

AEHS MATHEMA (Meta-Adaptation Technology Hypermedia for Electro-Magnetism Approach) combines the constructivist, socio-cultural and meta-cognitive teaching model and supports personalized and collaborative learning. The teaching strategies are based on [9] learning cycle and learning style model. The general purpose of the AEHS MATHEMA is to support learners through an interactive and constructivist educational environment in order to construct their knowledge in Physics and Technology, individually and/or collaboratively, and to overcome their misconceptions and/or learning difficulties

AEHS MATHEMA supports adaptive and intelligent techniques, such as curriculum sequencing, adaptive presentation, adaptive and meta-adaptive navigation, interactive problem solving, and adaptive group formation. More specifically it supports:

(1) adaptive presentation according to learner's learning style;

(2) four adaptive navigation techniques which are direct guidance (uses a "next" button), link hiding (hide non-relevant links), link annotation (use colors to indicate relevance), and link sorting (like in search engines);

(3) a meta-adaptive navigation technique in order for the learner selects the more appropriate navigation technique that best suits him or her;

(4) an interactive problem-solving method through personalized and collaborative problem-solving activities that exploits the teaching approaches of experimentation through simulations, guided inquiry and discovery, investigations and peer collaboration; and

(5) a method of adaptive group formation for peer matching.

The aim of this article is to highlight the main contributions of the architecture of the MATHEMA in the improvement of the functionality of AEHSs. The key idea is the decentralization of their functions.

The rest of the paper is organized as follows: In the section 2 related works for various models of architecture of AEHSs is presented. In the section 3 the architecture of the AEHS MATHEMA is presented in detail. In the section 3 a summary evaluation of the functions of the AEHS MATHEMA is presented and at the last section a summary is done and a discussion is taking place.

2. Related Work

Most of the Internet based systems use a variation of conditional reasoning where a decision is made to organize the text differently from fragments or select from a group of whole pages or even groups of pages. The link structure of a hyper-document can also be modified by color-coding, or sorting according to specific criteria based on student preferences or abilities. A student model is usually created

and maintained past the individual session and opened when a student logs into the system. This model retains in details all required information by the system from which the various decisions could be made. The decisions are to select between lessons or pages, which are going to be presented to that particular student.

For the AEHSs various models of architecture have been developed so far on which the designers of these systems are based on. In this section some models that are most suited to the MATHEMA system are presented.

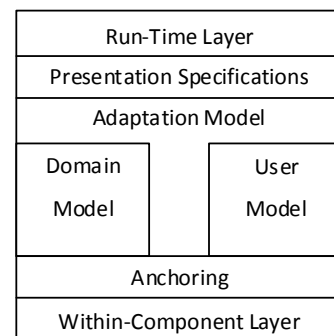The AHAM architecture model proposed by [6] is shown in the Figure 1.

Figure 1. AHAM reference architecture model.

It indicates that AEHSs must have three essential parts:

(1) The User Model that represents the learner's knowledge of the domain and his individual characteristics such as knowledge, preferences, experience, etc. This component allows extracting and expressing conclusions on the user characteristics. User Model allows changing several aspects of the system, in reply to certain characteristics of the user [3]. These characteristics represent the knowledge and preferences that the system assumes that the users or group of users have. A Student Model includes information referring to the specific knowledge that the system judges that the student possesses on the domain. Different AEHSs may store different information in a user model, besides a representation of the user's knowledge.

(2) The Domain Model that describes how the information is structured and linked together. It represents a set of domains concepts (teaching domain of the system). In different AEHSs these concepts can have distinct functions, weights and meanings. Most commonly, each concept is connected/related with other concepts. The most important function of this model is to provide a structure for the representation of the user domain knowledge. One approach is to store the estimate level of the user's knowledge for each concept. This value can be expressed quantitatively, qualitatively or in probabilistic form.

(3) The Adaptation Model or Teaching Model that consists of pedagogical rules which define how the domain model and the user model are combined to provide ways to perform the actual adaptation. An adaptive

engine performs the actual adaptation by adapting or dynamically generating the content of nodes and the destination and class of links in order to guide each individual user differently.

The AEHSs fit so nicely in this model. When designing a course it is important to first list the concepts. The next thing to do is to determine dependencies between the concepts. This leads to a structure of prerequisite relationships. The prerequisites are most common in educational material.

For each type of relationship (e.g., prerequisites) some generic adaptation rules can be defined. The rules together form the adaptation model in AHAM.

A key point in AHAM is that the adaptation rules are used to translate user actions into user model updates (as well as the presentation and adaptation of content). Also, the rules use the user model information together with the action information in order to determine the required user model updates.

In order to perform adaptation based on the domain model and user model is needed to specify how the user's knowledge influences the way in which the information from the domain model is to be presented. This is done by means of a teaching model which consists of pedagogical rules. The rules are used by an adaptive engine in order to generate the presentation specifications.

AEHSs applications need to maintain a permanent user model. Such permanent information which exists at the conceptual level also does not belong in the within-component layer because that layer deals with implementation-specific elements.

Today almost all AEHSs follow the general architecture of the AHAM reference model (but possibly with a different kind of rule or reasoning language and engine).

ADAPT [7] is an European Community funded project that aims to rectify the situation described in the introduction, by investigating current adaptive practices in various AEH environments and identifying the design patterns within them.

The ADAPT project has identified high level design dimensions for AEHSs, which are:

(1) context of use (CU),
(2) content domain (DM),
(3) instructional strategy (IS),
(4) instructional view (IV),
(5) learner model (LM),
(6) adaptation model (AM), and
(7) detection mechanism (DE).

These dimensions form the axes on which both an AEH problem and its solution can be represented.

[8] presented a similar architecture in the Proper system based on the AHAM model. Proper's architecture is a combined architecture of SCORM LMS and AEHS. As shown in simplified form, in the Figure 2, they adopt the typical SCORM Run Time Environment (RTE) structure adding an adaptation module and extending the preexistent Domain and User Models. Thus the prototype involves four main modules:

(1) The Domain Model (DM) that represents the domain knowledge of the system.
(2) The User Model (UM) that represents the particular user's knowledge of the domain as well as his individual characteristics; both these models comprise the Data Model of the system.
(3) The Adaptation Module (AM) which interacts with DM and UM in order to provide adaptive navigation to the course, and
(4) The RTE Sequencer that is triggered by course navigating controls and interacts with DM and delivers the appropriate educational content to the learner.

Educational content can be either SCO or Asset. Their system derives from SCORM 2004 Sample RTE Version 1.3.3 that is based on SCORM 2004 specification. Thus its architecture is a typical of a SCORM compliant LMS.
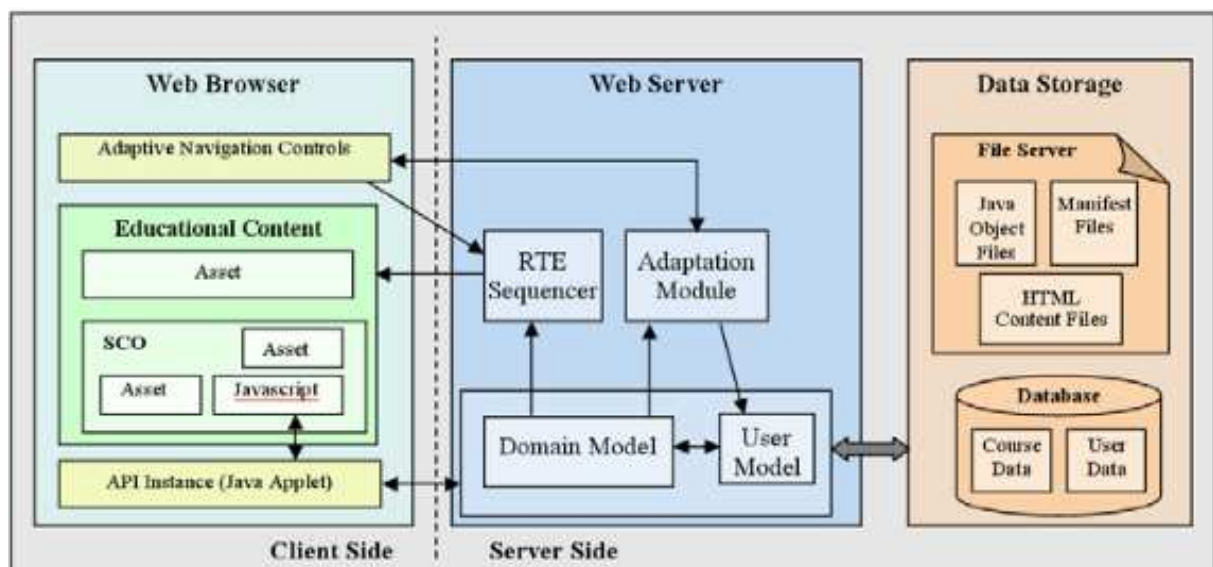

Figure 2. *Architecture of the Proper system.*

They use the Apache Tomcat 5.5 as Web and application server and the MySQL 5 as database server. System retrieve course files initially from a zip file, which contains a manifest xml file and all the html and media required files. The DM structure is exported by the manifest file and is stored into Java Object Files.

However additional data about the course is stored into the database. Moreover domain independent data of UM is stored into the database while at the same time, data about user knowledge is stored into Java Object Files.

The adaptive navigation techniques have been applied to it via Java servlets and Java Server Pages (JSP). All the runtime data about user actions and performance is stored into Java Object Files via JSP and Java servlets.

In the Figure 3 the architecture of the WELSA [14] system is presented. The adaptation component consists of a Java servlet which automatically generates the individualized Webpage, each time an HTTP request is received by the server. WELSA does not 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 Learning Object (LO) metadata, which 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 work-load 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.
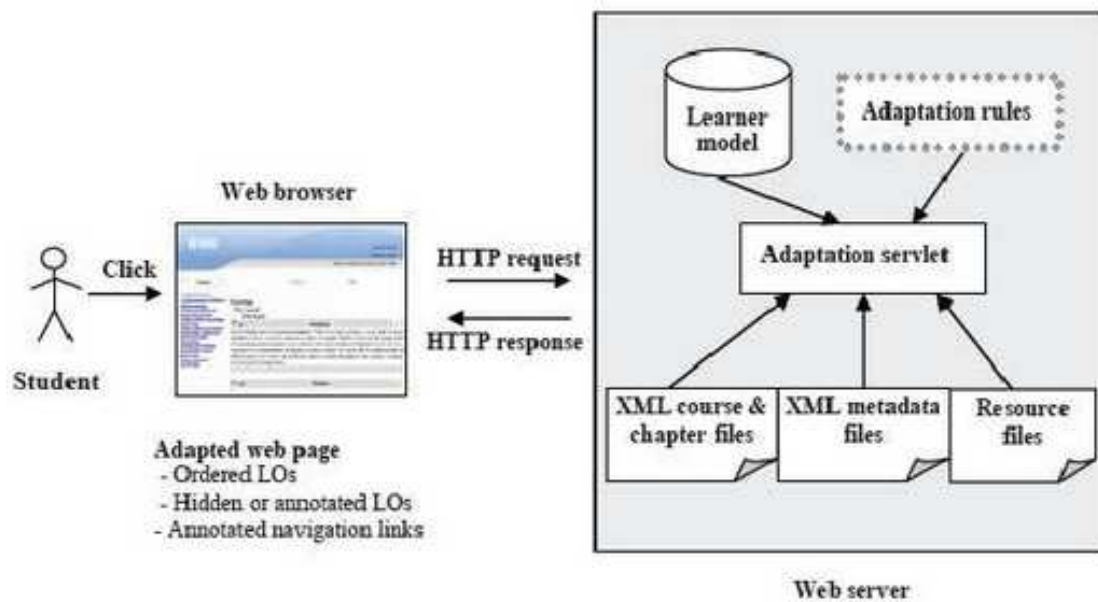

Figure 3. *Architecture of the WELSA system.*

[12] proposed a framework for AEHS (Fig. 4) based on JSP, Java servlets, and Javabeans. The Interaction Analyzer is responsible for acquiring information on learner's behavior. The Application Model represents the main features of the application in terms of a model of the domain, the instructional or learning theory used and how the domain concepts are grouped into learning units.

The Learner Model is the structure that contains the information on the learner's characteristics that allow the AEHS to adapt to these characteristics. The Adaptation Decision Model is responsible for deciding what the system should do in terms of presentation and navigation adaptation on the basis of the conclusions draw by the Interaction Analyzer, parameters from the Learner Model and

information from the Application Model.

The Presentation Generator is responsible for generating what will be presented to the learner as a result of processing the information received from the Adaptation Decision Model and Application Model. The Presentation Generator requests a composition of the presentation to the JavaServer Pages.

As JavaBeans are components of an application in the context of JavaServer Pages or servlets, they are suitable to implement the Presentation Generator and the Adaptation Decision Model. Additionally, the JavaBeans offer advantages of separating the programming logics from presentation, as for the Presentation Generator and the final page composition.
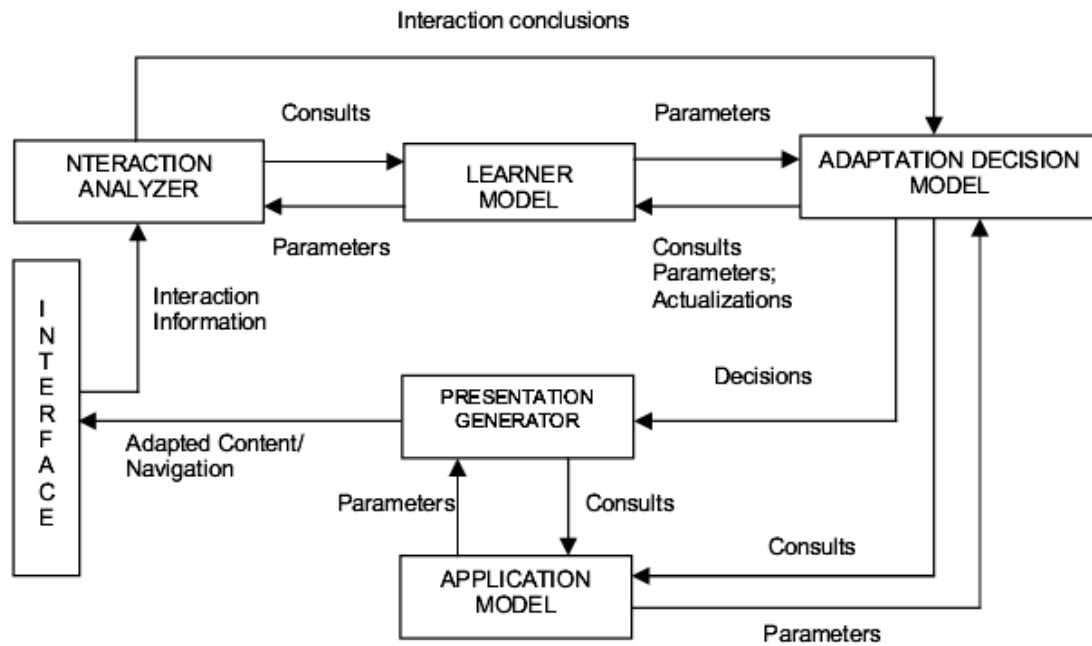
Interaction conclusions

Figure 4. Oliveira and Fernandes framework for AEHSs.

ULUL-ILM [1] is an AEHS that focuses on student's learning styles. The purpose of ULUL-ILM is to provide the AEHS that can recognize student' learning style automatically in real-time and then presents the learning content adaptively based on student learning style. It enables to recognize the student's learning style automatically in real time by means of Multi Layer Feed-Forward Neural network (MLFF). The MLFF is embedded to the system because of its ability to generalize and learn from specific examples, ability to be quickly updated with extra parameters, and speed in execution, making it suitable for real time applications. The system then enables to present and recommends a variety of learning contents adaptively towards each of the student's learning style identified in the student model through the adaptation model. The system then analyzes the learning content on each of the learning material, and then comes up with the generated teaching strategies by means of the teaching strategy generator and fragment sorting. The result of that analysis is called domain model. The adaptation model enables the system to adaptively presents the content, based on the student's learning style by combining the fragment sorting and adaptive annotation technique. The course player in ULUL-ILM enables the system to adaptively presents the content with various teaching strategies towards each of student's learning style.

3. Architecture of the AEHS MATHEMA

AEHS MATHEMA architecture is based on Web technologies similar to client/server distributed computing architecture of MIT and it is mainly supported by Apache Tomcat Server 5.5 and MySQL 6.0 relational database server.

For supporting features, it uses the (JSP) pages, Java servlets, JavaBeans, JavaScripts, and Apache Struts technologies. Apache is a server of choice because of its open standard and source-code, which provide easy connectivity to Java servlets. MySQL is the database of choice since there is extensive experience on Campus and provides the ability to implement adaptation rules. The content loaded to the MySQL database is accessed via JDBC API. This is an interface for Java that standardizes database connections, queries, and results of relational databases.

Java servlet technology and JSP pages are server-side technologies that have dominated the server-side Java technology market; they have become the standard way to develop Web applications. A servlet is a Java class that implements the Java servlet API, a protocol to respond to HTTP requests.

These servlets are complete programs that are capable of creating JSPs. A servlet allows a programmer to utilize whatever functions a programmer needs including conditional branching and loops. This allows for much more flexibility in creating the page than XML. Oddly enough, servlets do not face any of the problems faced by classical CGI programming because a servlet has a lifecycle. Even a loss of connection may not kill it, unless the Server side program kills it. It is light weight, which means that it does not require a large amount of processing power on creation of the servlet instance, and it accesses databases using JDBC which offers a secure way of accessing many well-established database brands.

Servlets are capable of eliciting student responses on more than one question and analyzing them to find out the strengths and weaknesses of that student to direct them towards remediation. Servlets dynamically create JSP pages according to student requirements.

JavaBeans technology comes into play. This technology is a portable, platform-independent component model that lets developers write components and reuse them everywhere. In the context of JSP pages, JavaBeans components contain business logic that returns data to a script on a JSP page, which in turn formats the data returned from the JavaBeans component for display by the browser. A JSP page uses a JavaBeans component by setting and getting the properties that it provides. The benefits of using JavaBeans components to augment JSP pages are the following [10]:

(1) Reusable components: Different applications will be able to reuse the components.
(2) Separation of business logic and presentation logic: You can change the way data is displayed without

affecting business logic. In other words, web page designers can focus on presentation and Java developers can focus on business logic.

(3) Protects your intellectual property by keeping source code secure.

The Apache Struts is a formalized framework for the architecture style design pattern Model View Controller (MVC). This framework is best used for complex applications where a single request or form submission can result in substantially different-looking results.

Figure 5 shows the architecture of the AEHS MATHEMA. It shows how all units work together to make the system operational. Bellow the operation of the units will be presented and analyzed.

Figure 5. Architecture of the AEHS MATHEMA.

3.1. Didactic Model

The didactic model uses the information contained in the

learner model to provide the two basic services, namely, adaptive presentation and adaptive navigation. It is a set of pedagogical rules that combine the learner's model with the domain knowledge for adaptive performance. In the field of

selecting of the most appropriate teaching strategy, the learner's learning style is taken into account during the learner's study.

The didactic design of the AEHS MATHEMA supports learners in constructing knowledge, choosing and achieving their learning goals, recognizing what they have already learned and what they are capable to do, and judging personal progress of their learning. It also offers appropriate teaching strategies to match students' learning styles, develop critical thinking and self-regulation, as well as collaborative activities, encouraging them to actively participate.

Moreover, offers facilities in recognizing learners' misconceptions and assisting them in self-correction through reflection, and providing multiple representations of learning.

3.2. Domain Model

A component of an adaptive educational system is the representation of knowledge.

The domain knowledge is structured in a way that supports the ability of the system to choose the educational material, depending on the learner's requirements and current status.

The domain knowledge of the AEHS MATHEMA is the basis of the system's adaptation. During the design, it was taken seriously into account that the representation of knowledge, as well as the quality of the educational material significantly influences the quality of educational result.

The system was designed to include a set of predefined cognitive goals, from which each learner is required to choose. Each cognitive goal relates to a set of concepts of the subject, which are necessary for its understanding.

In the AEHS MATHEMA, the hierarchical representation of the domain model is adopted by the ELM-ART [5], as follows:

The first level defines the cognitive objectives, the second the basic concepts and the third the pages with the corresponding educational material.

It also defines the relationships between the concepts (prerequisites) and the level of difficulty of the concepts (see part of them in the Table 1). Content is on separate pages, enabling it to be re-used.

3.3. Student Model or Learner Model

Student model contains information about student characteristics that allows the system to make adjustments using these characteristics. The model that supports the AEHS MATHEMA is the overlay model. It maintains the model of each learner, working in the system, and updates it throughout the duration of the interaction.

The characteristics that the system maintains in learner model for each learner are: name, gender, username, password, learning style, level of knowledge for each cognitive goal (general performance), level of knowledge for each basic meaning of the cognitive goal (quantitative and qualitative characterization), concepts that have been successfully evaluated for each cognitive goal, current page, current teaching method, current navigation technique, navigation history, meta-adaptive navigation status, initial Web experience and knowledge level (pre-existing knowledge) in the current cognitive goal, ways of presenting feedback messages, preference for assistance by Advisor in terms of its navigation, course sequence, and phases of the guided dialogues in interactive problem solving process in which each learner participates.

Table 1. Degree of difficulty and prerequisites of basic concepts.

a / a	Basic concepts	Degree of difficulty	Prerequisites
1	Electric Field Intensity	1	-
2	Electric Flow	2	1
3	Gauss's Law	3	1,2
4	Electric Potential - Electric Potential Difference	1	1,2
5	Charged Particle Movement into Homogeneous Electrostatic Field	2	1,4
6	Capacitor and Capacity	1	1,4

The Student Model is transparent to the learner (open learner model) and is verifiable by the learner, that is, the learner can access his/her model and change the content of some characteristics.

The learner during the study selects the cognitive goal he or she wishes to study, following his or her personal course in the educational content, exploits the system's assistance, navigating and studying, is informed about the information that the system maintains, who can change them by intervening and directing the system's adaptation in this way,

in terms of curriculum sequencing, adaptive presentation and navigation and adaptive group formation, selects his or her learning style, selects the navigation technique, activates and deactivates the curriculum sequencing, selects the way that the feedback messages are presented, changes his or her knowledge level for each basic concept of the cognitive goal, and activates-deactivates the Advisor.

Figure 6 shows a snapshot of the page showing the characteristics of the learner model where the learner can be informed about them but also modify them.

Figure 6. A snapshot of the characteristics of the Student Model.

3.4. Adaptation Engine

The adaptation engine decides on adaptive presentation and navigation in relation to Student, Domain and Didactic models. It also takes into account the decisions of the data collection and monitoring unit of the learner's actions. The system, during the learner's study, monitors his or her interactions with the system and his or her assessment and accordingly updates the links to the course material provided and/or adapts learning material to learner according to his or her learning style. The adaptive navigation techniques that it supports are: direct guidance, link hiding, link annotation, and link sorting.

3.5. Meta-Adaptation Engine

This is an innovation on the architecture of AEHSs. It is the second level meta-adaptation engine that is responsible for monitoring the cognitive improvement of the learner in the cognitive goal that he or she studies, after *n* successful evaluations of his or her knowledge in basic concepts, showing him or her advantages and disadvantages of the four navigation techniques that it supports. It also offers additional information about the navigation techniques that it supports.

In AEHS MATHEMA, meta-adaptive navigation works as follows: The first time that the learner enters the system, he or she is asked to state his or her Web experience and level of knowledge in the cognitive goal he chose to study (pre-existing knowledge).

Following the learner's statement, the system suggests the most appropriate navigation technique to him or her, taking into account the level of his or her Web experience and his or her level of knowledge in the subject he or she chose to study as follows:

(1) Direct guidance: Little or no Web experience and little or no knowledge of the cognitive goal.
(2) Link annotation in conjunction with curriculum sequencing: Little or no Web experience and modest knowledge of the cognitive goal.
(3) Hiding links: A lot of Web experience and little or no knowledge of the cognitive goal.
(4) Link annotation: A lot of Web experience and modest knowledge of the cognitive goal.

The meta-adaptation engine proposes to the learner to change the navigation technique if he or she wishes after *n* successful assessments of the basic concepts of a cognitive goal he or she studied, showing him or her the pros and cons and additional information on each of four navigation

techniques, and the learner decides whether to change it or not.

Figure 7 shows a snapshot of the meta-adaptation result.

The learner has the ability to change his or her navigation technique.

Figure 7. A snapshot of a meta-adaptation result.

3.6. Application Module

This module is responsible for transferring the training material to the presentation unit, for giving feedback, for creating guided dialogues and for evaluating the learner knowledge. It includes the content of the JSP pages, Java servlets responsible for feedback and creation of guided dialogs in problem-solving, Java servlets for creating evaluation questionnaires, and so on. In this design phase, the system does not follow a standard metadata description but the system supports data transfer objects.

3.7. Presentation Module

It is responsible for what will be presented to the learner as a result of the processing of the information that arrives from

the adaptation engine, the meta-adaptation engine, adaptive group formation module, and application module.

The pages displayed to the learner are dynamically generated. In this module there is the basic servlet, which deals with the presentation of the training material and other JSP pages and servlets, such as the page and the corresponding servlet to inform the learner about the characteristics of all learning styles, the servlet for the curriculum sequencing, the servlet for selecting and presenting the concept that the student selects to study according to his or her learning style, etc.

Figure 8 shows the user interface, as seen by each user on his or her remote computer, sent to him or her by the presentation module.
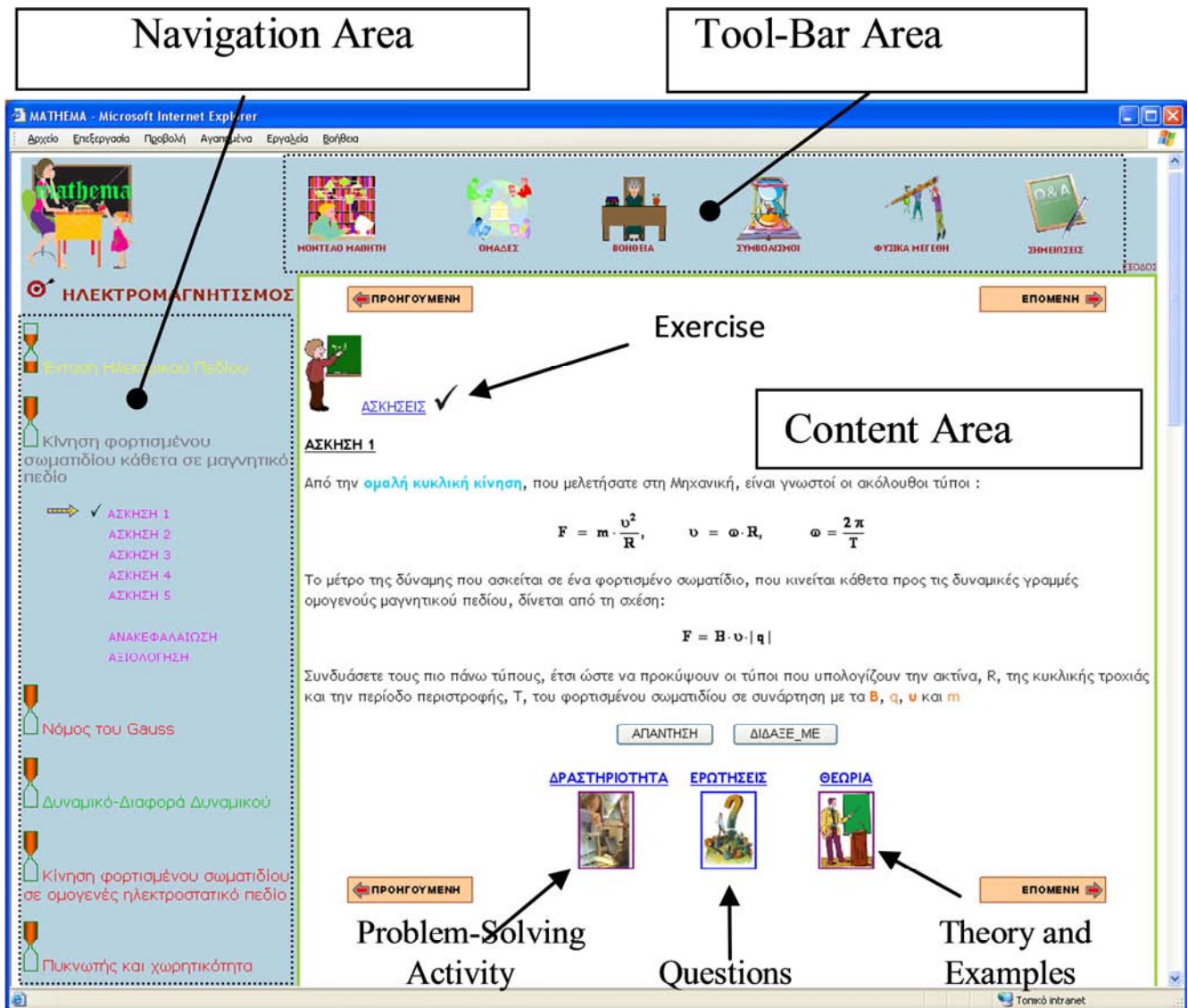
Figure 8. The user interface of the AEHS MATHEMA.

3.8. Module of Data Collection and Monitoring of Learner's Actions

This module is responsible for obtaining information about the data the learner entries in the system and for monitoring his or her actions (interaction with the system). More specifically it:

(1) initializes and updates the Student Model;
(2) collects and stores the data that the learner answers to open-ended questions, entries in tables, registers in notes, etc;
(3) monitors the interactions between the learner and the system to get information about his or her choices in terms of the questionnaires, the phases of guided dialogs, the links, the options for meta-adaptive navigation, the visited pages, and more.

Figure 9 shows a snapshot of the page responsible for selecting preferences of the learner.

Figure 9. *Student preference initializations for the Learner Model.*

3.9. Adaptive Group Formation Module

MATHEMA supports the learner to find the most suitable peer for the formation of a collaboration team.

Following a study [13] conducted on the formation adaptive groups in the MATHEMA, it was decided that: when the system creates a priority list of candidate peers for an interested learner, in the first and second position of the priority list, the algorithm will place the candidate collaborators with a concrete or abstract style and with the same active or reflective dimension of their learning style as follows: If the learner has a concrete learning style, then the algorithm will place the candidate collaborators with a concrete learning style in the first position, and in the second

position, the candidate collaborators with an abstract learning style. If the learner has an abstract learning style, then the algorithm will place the candidate collaborators with an abstract learning style in the first position, and in the second position, the candidate collaborators with a concrete learning style. In candidate collaborators belonging in the same position, the classification is according to their level of knowledge in the current cognitive goal up to that moment.

After creating the list, the system informs the learner that his or her most important candidate collaborator is at the top of the list, while the less important is at the end of the list.

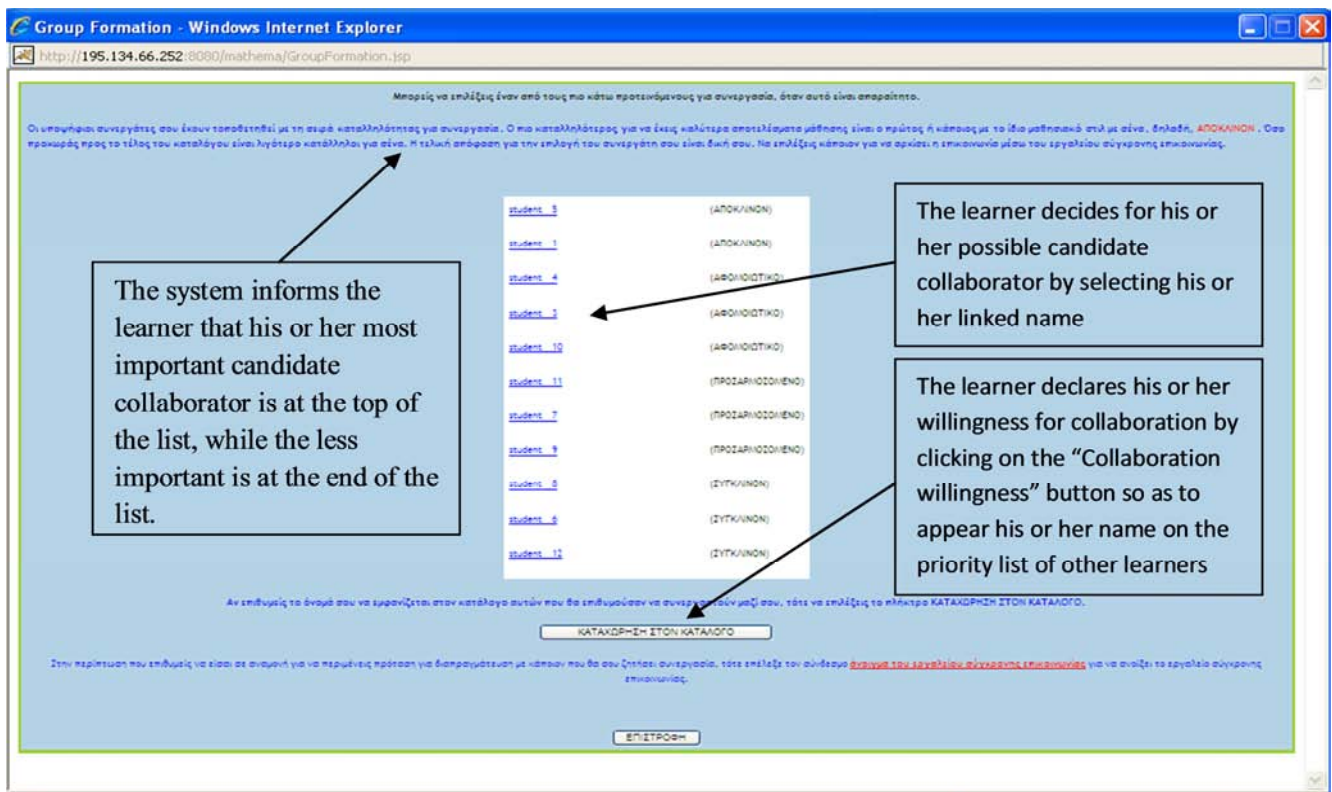Figure 10 shows a snapshot of adaptive group formation in the MATHEMA.

Figure 10. *A snapshot of adaptive group formation in the MATHEMA.*

3.10. Synchronous Communication Tool

This module is responsible for monitoring and supporting synchronous communication between learners via a chat-tool. The collaboration protocol that uses the synchronous communication tool is as follows:

(1) The learner declares willingness to collaborate either by selecting his or her partner from the priority list of candidate collaborators or by declaring a desire for collaboration so that others who would like to work with him or her can choose it, while activating the synchronous communication tool.

(2) Learners negotiate for collaboration and, if they come in agreement with each other, state it in the system to not include them in the priority list of other learners who would like to work with them. Also, when a learner-to-learner connection is made through the synchronous communication tool, then no other learner is allowed to join.

(3) After negotiating and updating the system that they come in agreement with each other to collaborate, learners can start collaboration.

The Figure 11 shows a snapshot of a dialog between the learners Giannis and Mary taking place via the Synchronous Communication Tool (chat-tool).

4. Evaluation of the Functions of the AEHS MATHEMA

The evaluation of an educational tool is important and it must be ensured that the correct methods and techniques are used. Assessment of the pedagogical and technological aspects of the AEHS MATHEMA has been carried out.

Research has shown that high school students increase their performance by studying through the AEHS MATHEMA [13].

The evaluation of the system was carried out by students of the Department of Informatics and Telecommunications of the University of Athens, Greece. Generally, the students felt that almost all of the system's functions are quite useful (83.7 to 100 percent) and easy to use (58.1 to 93 percent).

5. Results and Discussion

Regarding adaptive and intelligent techniques of the AEHS MATHEMA mentioned above, the students having evaluated them consider that the adaptive presentation of different teaching strategies is quite useful (93.0 percent) and fairly easy to use (86.0 percent), the different ways of navigating are quite useful (100.0 percent) and handy (90.7 percent), the meta-adaptive navigation is quite useful (88.4 percent) and fairly easy to use (86.0 percent), the adaptive group formation of collaborative teams to assist them in collaborative problem solving is useful (83.7 percent), but less user-friendly (58.1 percent).
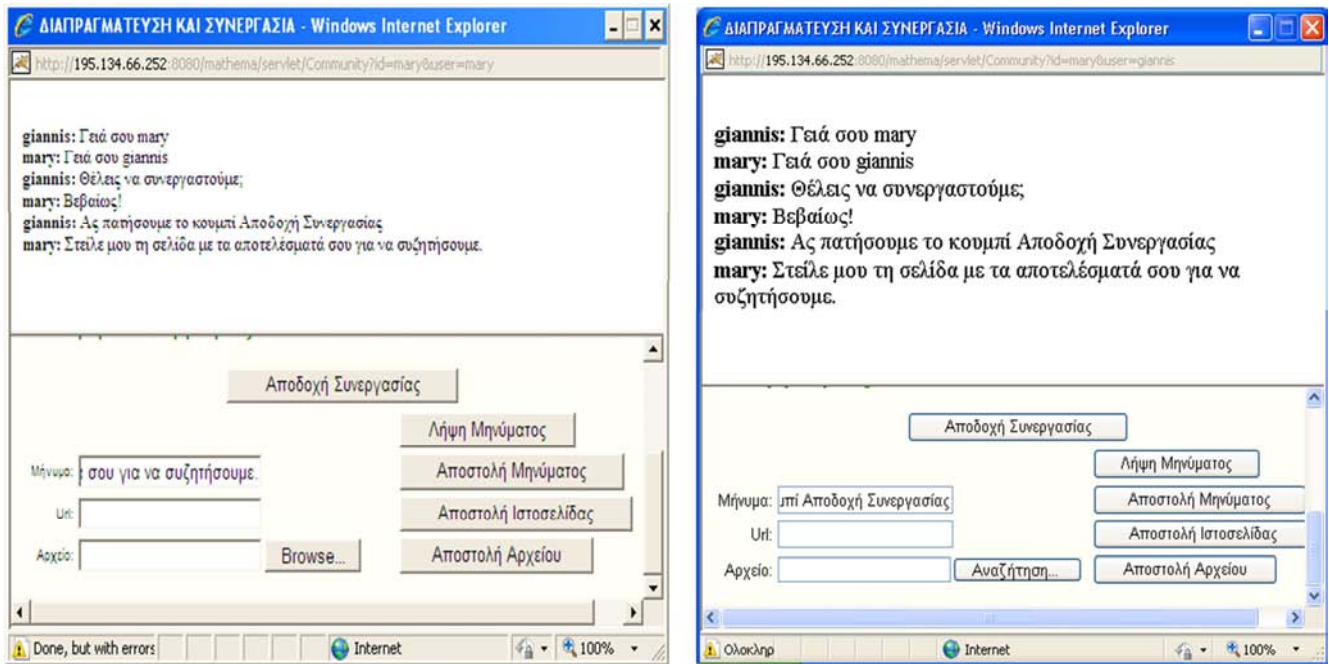
Figure 11. Synchronous Communication Tool used by the MATHEMA.

asynchronous communication tool, have been improved.

6. Conclusions

This article presents the trends of architecture of AEHSs so far and then an extensive description of the architecture of the AEHS MATHEMA. The implemented AEHSs so far use various techniques to implement their functions. Most of AEHSs presented in this paper are based on JSP, Java servlets, and Javabeans technologies to implement their functions, as implemented by AEHS MATHEMA. Furthermore, the presented AEHSs above have at least three main modules, such as Student Model, Domain Model, and Adaptation Model as the AEHS MATHEMA also has. The architecture of AEHSs becomes more complex as more and more functions are implemented. MATHEMA supports adaptive and intelligent techniques, such as curriculum sequencing, adaptive presentation, adaptive and meta-adaptive navigation, interactive problem solving, and adaptive group formation. In order to support all these functions the architecture of the MATHEMA is more complicated from other AEHSs. The key idea is the decentralization of their functions. In this paper the operation of each module of the system is briefly described which contributes to the required functionality of the system.

The main contributions of the architecture of the MATHEMA in the improvement of functionality of AEHSs are the meta-adaptation technique, the adaptive group formation technique, the problem-solving technique, and the synchronous communication protocol that it supports.

The evaluation of the AEHS MATHEMA was encouraging and rewarding since the evaluators considered that almost all system functions are quite useful and easy to use. Taking into account the observations and recommendations of the evaluators, some system functions, such as this offered by the

References

[1] Bayasut, B. L., Gede, P., & Halizah, B. (2013). ULUL-ILM: The Design of Web-Based Adaptive Educational Hypermedia System Based on Learning Style. In *13ᵗʰ International Conference on Intelligent Systems Design and Applications (ISDA)*. Kuala Lumpur: IEEE.

[2] Brown, E., Cristea, A., Stewart, C., & Brailsford, T. (2005). Patterns in Authoring of Adaptive Educational Hypermedia: A Taxonomy of Learning Styles. *Educational Technology & Society*, 8 (3), 77-90.

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

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

[5] Brusilovsky, P., Schwarz, E., & Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web*, Proc. Third Int'l Conf. Intelligent Tutoring Systems. (ITS '96)*, C. Frasson, G. Gauthier & A. Lesgold, eds., vol. 1086, pp. 261-269.

[6] de Bra, P., Houben, G. J., & Wu, H. (1999). AHAM: A dexter-based reference model for adaptive hypermedia. *In Proceedings of the ACM Conference on Hypertext and Hypermedia*, Darmstadt, Germany 147–156.

[7] Garzotto, F., & Cristea, A. I. (2004). ADAPT: Major design dimensions for educational adaptive hypermedia. *Paper presented at the ED-MEDIA 2004 Conference*, June 21-26, 2004, Lugano, Switzerland.

[8] Kazanidis, I, & Satratzemi, M. (2009). Adaptivity in ProPer: An Adaptive SCORM Compliant LMS, International Journal of Distance Education Technologies (IJDET). 7 (2), pp. 44-62.

[9]   Kolb, D. A. (1984). *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Clifs, N. J.: Prentice-Hall Inc.

[10] Mahmoud, Q. H. (2003). Servlets and JSP Pages: Best Practices. Available from: http://www.oracle.com/technetwork/articles/javase/servlets-jsp-140445.html.

[11] Martins, A. C., Faria, L., Vaz de Carvalho, C., & Carrapatoso, E. (2008). User Modeling in Adaptive Educational Hypermedia Systems. *Educational Technology & Society*, 11 (1), 194-207.

[12] Oliveira, J. M. P., & Fernandes, C. T. (2003). A Framework for Adaptive Web-based Educational Systems. In: *Workshop on Applications, Products and Services of Web-based Support Systems in conjunction with the IEEE International Conference on Web Intelligence*, Halifax, Canada, pp. 55-62.

[13] Papadimitriou, A., & Gyftodimos, G. (2007). Use of Kolb's learning cycle through an adaptive educational hypermedia system for a constructivist approach of electromagnetism. *In Proceedings of 4th* WSEAS/IASME *International* Conference on Engineering Education, (pp. 226-231), Agios Nikolaos, Crete Island, Greece.

[14] Popescu, E., Badica, C. & Moraret, L. (2010). Accommodating Learning Styles in an Adaptive Educational System. *Informatica,* 34, pp. 451–462.

[15] Somyürek, S. (2015). The New Trends in Adaptive Educational Hypermedia Systems. *International Review of Research in Open and Distributed Learning*, 16 (1), pp. 221-241.