

Dynamic User Interface Adaptation Driven by Physiological Parameters to Support Learning

Giuseppe Ghiani, Marco Manca, Fabio Paternò

CNR-ISTI, HIIS Laboratory

Via Moruzzi 1, 56124 Pisa, Italy

{giuseppe.ghiani, marco.manca, fabio.paterno}@isti.cnr.it

ABSTRACT

Technology to make physiological measurements related to attention and cognitive load is becoming more affordable. We propose a solution based on combining the exploitation of dynamic user information gathered through such technology with a rule-based strategy for adaptation of e-learning Web applications. We focus on users' physiological data and aspects relevant for the task being carried out. A flexible rule-based approach allows designers and developers to define a wide range of rule compositions to express changes in the user interface based on how the user feels and behaves. The overall goal of the framework is to serve as a tool for content developers of Web applications, such as operators of online Learning Management Systems, and for their end-users. In this domain, through our approach teachers can create their educational contents, and specify how they should dynamically adapt to students' behaviour in order to improve the learning process.

Author Keywords

Adaptation, Web, Learning, Physiological parameters.

ACM Classification Keywords

H.5 Information Interfaces and Presentation; H.5.2 User Interfaces.

INTRODUCTION

Adaptation has been considered in various proposals in recent years (e.g. [1, 2]). However, its potential has not yet been fully exploited. In this work we focus on adaptation for e-learning applications, which is important because there can be many possible learning styles. For example, some people 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. In addition, such preferences are not static but the same people can change them according to their personal state.

In order to address such issues we propose a software architecture and its prototypical implementation able to exploit information gathered through various sensors related to some psycho-physiological parameters. The solution proposed is for Web environments, which are the most widely used in e-learning applications.

Various approaches to identifying different learning styles have been proposed, which are based on questionnaires answered by the students before application access, or on server log files in order to monitor resources that have been accessed by the students, their frequency of use and time spent on them [2]. We aim to provide real time detection of a user's state, and consequent adaptation to the detected engagement. For this purpose, we consider parameters such as users' brain-activity data (attention/meditation) while users interact with e-learning applications. In this manner, we can predict the level of user engagement (focus/relaxation) with the interface, and detect and monitor the students' preferred perceptual mode for learning information. In addition, we also consider gaze behaviour as another interesting indicator of the level and focus of user attention, which can also be a good source of information for identifying the elements that attract more interest or are problematic.

In general, any parameter characterizing users and the situation where they are operating can be considered relevant to obtain user interface adaptation in order to better support user tasks. Regarding adaptive user interfaces however relevant information is limited to what can affect the current user activity [5]. In our investigation, to provide real time adaptation based on the mental state, we thus focus on contextual parameters that arise from user behaviour monitoring. In order to enable adapted user interfaces for various devices, a crowd-based approach to adaptation has been proposed [7], while we propose a platform able to support adaptive user interfaces obtained through user models that are updated by monitoring physiological parameters.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
EICS'15, June 23 - 26, 2015, Duisburg, Germany
© 2015 ACM. ISBN 978-1-4503-3646-8/15/06...\$15.00
DOI: <http://dx.doi.org/10.1145/2774225.2775081>

As stated in [5], the job of adaptive application programmers is basically to create mappings between events and transformations. This is done according to the “event-control-transformation” schema, consistent with the “event-condition-action” paradigm that we adopt.

Context-dependent adaptation of Web applications is tackled by previous work, for example in [4] automatic multimodal adaptation is provided for mobile users through the use of a model-based framework. However, the diverse contextual aspects considered were mostly related to the environment (e.g. noise level, user’s location) and the adaptation implied always a change of interaction modality either in the input or output. In the proposed solution we instead focus on user physiological aspects in stationary settings and, instead of applying a model-based framework, we directly update or add/remove only specific parts of the Web page without the need for a model-based infrastructure. We believe that this approach is more suitable with the scenario considered because high variability of user’s physiological parameters can trigger frequent adaptations, and updating page elements client-side via JavaScript is more performing than doing model-based reverse engineering server-side.

Previous work [3] showed that adaptation of e-courses leads to better results. They proposed a method for course adaptation based on adapting presentation of content, communication methods, and organization of online activities to students’ learning styles and preferences. Analyses of e-learning system adaptivity showed that cognitive characteristics of students, such as learning style, are of the greatest importance for successful adaptation. However, their work was not able to address real time adaptation. Therefore, it was necessary for teachers to monitor students’ progress and move them to another cluster, if necessary. Our solution provides a step forward filling in this gap with its ability to change presentation, navigation, and content according to the actual user behaviour detected through different physiological parameters. In addition our solution has a good level of generality because it can exploit in modular ways on the one hand various types of sensors and on the other hand various e-learning applications. Thus, it is more general than solutions that just connect a sensor with an application to obtain ad-hoc adaptation as in Bravo [6].

Other contributions have addressed adaptive educational systems but, in some cases, they required a pre-processing for analysing the history of the interactions on the graphical interface of the students, thus they were not able to provide real-time adaptivity. For example, in [8] the type of adaptations supported were limited to adaptive sorting and adaptive annotation techniques.

To summarise, the main goal of this work is to present a software architecture, and its reference application, for supporting dynamic adaptation of Web applications based on physiological data in the learning domain. The

adaptation is specified through rules structured in terms of events, conditions, and actions. Thus, it is sufficient to change the rules to obtain different adaptive behaviours. We also report on an example application and an associated user test. In the paper, we introduce the architecture of the proposed solution, then we describe an example application in the e-learning domain for which some adaptation rules based on dynamic physiological data have been specified, and we report on a first user test. Lastly, some conclusions and indications for future work are provided.

THE SOLUTION ARCHITECTURE

The solution can support various possibilities in terms of change type, granularity, and user interface aspect. Indeed, any type of change in the UI elements can be addressed (we can add, remove, and change elements). In terms of granularity, the changes can range from very detailed aspects (such as an attribute of one HTML element) to modifying the entire application. The changes can involve the presentation (layout, graphical attributes, styles), the navigation (which can be more or less sequential or with the possibility to have more or less links to related contents), and the content, which can be more or less detailed or with multimedia support depending on the characteristics of the user and the current mental state.

In the deployment phase the adaptation software architecture has two main inputs: the learning applications and the adaptation rules. At run-time, first the e-learning application is loaded (1 in Figure 1). Then, the application subscribes to the adaptation engine (2), which maintains a table associating the relevant adaptation rules with each application. The adaptation rules are specified through an XML-based language, and are composed of three parts: the events that trigger them, an optional part dedicated to specific state conditions, which should be checked as well in order to actually activate them, and the action part that indicates what changes will take place. The adaptation engine subscribes to the user model manager in order to be informed when the relevant events and conditions occur (3). Such user model manager is responsible for detecting the dynamic events occurrences along with checking the relevant conditions every time the event parameters values are updated (e.g. when the user attention level changes). For this purpose, the user model manager receives user-related information from the peripherals (such as the eye tracker, the brain waves biosensor, etc.). Such data is actually sent to the user model manager by dedicated pieces of software, which is connected to the peripherals, elaborates the data and sends them few times per second to the user model manager (4). When one or more of the subscribed events and conditions occur, the user model manager notifies the adaptation engine (5). The adaptation engine is aware whether one or more notification is associated to the triggering of adaptation rules concerning the current application (because it hosts an association table for this purpose). When a rule is triggered, the changes associated

to the corresponding actions are transmitted (6) in JSON format to a script injected in the e-learning application. The main aim of this script is to identify the DOM elements that have to be modified client side (7) according to the changes specified in the adaptation rules. The script has been made general enough to be easily injected in any existing Web application.

Injection consists of linking the script and some references to support libraries on the existing Web page that is being enhanced with adaptive capabilities.

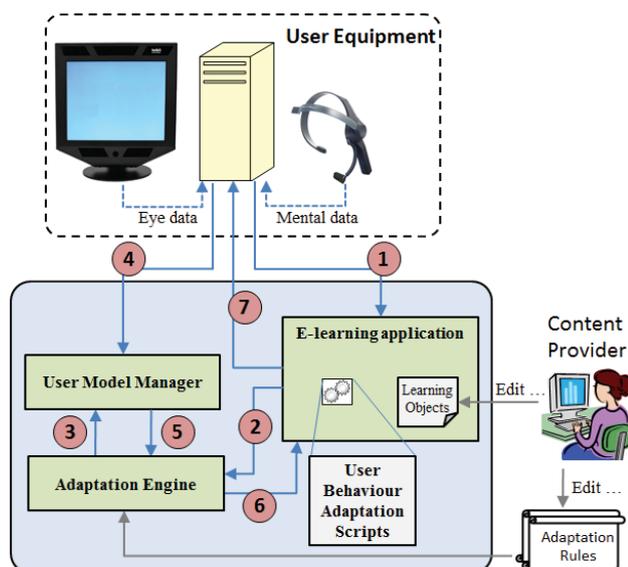


Figure 1: The proposed system architecture.

It is worth noting that this procedure can be done independently of the document type and its degree of dynamicity. For instance, the script can be included in a static HTML page as well as in a PHP file.

In detail, the functionalities of the injected script are: (i) Subscribing to the adaptation engine; (ii) Getting the current observed point (x and y coordinates received from the user model manager) in the display and mapping it into the application window coordinates; (iii) Identifying the gaze point paragraph and informing the user model manager; (iv) Setting an attribute to the currently observed paragraph; (v) Waiting for the actions corresponding to one or more event occurrences from the adaptation engine; (vi) Prompting the user about the potential triggering of adaptation rules and apply each action to the UI.

THE ADAPTATION RULES

The language for specifying the adaptation rules has various features in order to obtain sufficient expressiveness to support the various possible cases. It includes an “else” branch expressing the actions to do when the associated condition is not verified. Events can be composed through Boolean operators (AND, OR, NOT, XOR, sequence) and

time constraints can be declared to restrict the interval of event occurrence. Actions can refer to an element of the Web page to update in absolute or relative manner (e.g. with respect to the previous or next element)

The event part of an adaptation rule can involve a single event or an event composition, e.g. a sequence of subsequent events. One or more conditions can be expressed for the parameters variation that characterize the event. These range from simple conditions (e.g. attention level lower than 20), to composite ones (e.g. attention above 60 and below 100). Conditions can even comprise more than one attribute, such as the attention level and the amount of clicks in the current session.

THE ACQUISITION OF PHYSIOLOGICAL DATA

To monitor the parameters of the user behavior, we have exploited two peripherals: a Brain-Computer Interface (BCI) and an eye tracker.

As BCI we have chosen a Neurosky¹ device (in the following referred to as biosensor), a Bluetooth-enabled battery-operated peripheral that capture user’s EEG signals and can easily worn as an headset. We have opted for the Neurosky biosensor since it is less invasive and easier to wear than other devices (such as Emotiv², which requires the use of many wet sensors around the scalp). Neurosky uses only one dry sensor in the front and an ear contact point. It is thus no more invasive than a traditional headset.

NeuroSky devices embed a technology called ThinkGear³, it includes the sensor that touches the forehead, the contact on the ear pad and a chip that processes and elaborates all the collected brainwaves and return the eSense Meters (Attention and Meditation).

We have implemented a Java application that connects to the biosensor via Bluetooth and, by exploiting the Neurosky API, retrieves the current attention and meditation level, as well as any eye blink events. The biosensor is indeed able to monitor such parameters by relying on electroencephalography (EEG) and electromyography (EMG).

For tracking user gaze, we have exploited a Tobii⁴ Eye Tracker 1750. The Tobii SDK furnishes an API for retrieving the currently observed point with respect to the screen coordinates. We have implemented a C# application that gets the current coordinates and sends them to the user model manager. Figure 2 shows the resulting working environment with the peripherals used for our study. As can be seen, they allow users to interact with the application without hindering their movements.

¹ <http://neurosky.com/products-markets/eeg-biosensors/>

² <https://emotiv.com/>

³ http://developer.neurosky.com/docs/doku.php?id=thinkgear_connector_tgc

⁴ <http://www.tobii.com>



Figure 2: A user wearing the Neurosky headset and using the Tobii Eye Tracker.

EXAMPLE APPLICATION

We have integrated our adaptation platform with the Moodle⁵ Learning Management System (LMS). We have chosen it because it is an open source Web environment that educators can use to create effective online learning sites, and it is considered the most popular LMS⁶. However, our approach can be applied potentially to any existing Web-based LMS.

We have injected our adaptation scripts into one of the main Moodle PHP files, and used a course on “Computer Revolutionary Invention”, which retraces the history of computer science from the first mechanical computer until today (see a part in Figure 3).

In the example application we considered the following parameters of the user model:

- Current screen coordinates observed by the user (detected by the eye tracker);
- Total time that the user has spent looking towards the display, but outside its borders in the last three minutes (detected by analyzing the history of eye tracker out-of-screen coordinates);
- Number of eye blinks in the last three minutes (blinks are detected by Neurosky biosensor);
- Currently observed paragraph element (each titled section of the document is considered to be a paragraph), which is highlighted in real time with a green border;
- Amount of time during which the user has observed a certain paragraph, filtered to avoid that short fixations

outside the paragraph reset the timer (i.e. the current paragraph timer is reset only if the user gaze is averted for more than 4 seconds);

- Mean attention level, i.e. how much the user is focused, that is obtained by averaging the last 10 samples from the Neurosky biosensor. Averaging allows filtering the samples from occasional peaks.

id	Trigger	Adaptation
A	Long Total Time Out Screen	Display all the simplified contents.
A'	Low Total Time Out Screen	Hide any simplified content.
B	Lost Attention	Display simplify version for watched paragraph.
B'	Gained Attention	(Same as A')
C	Long Time Element High Blink	(Same as B)
C'	Low Blink	(Same as A')

Table 1: The six adaptation rules.

We wrote 3 pairs of complementary rules, in the following indicated as A/A', B/B' and C/C' (as shown in Table 1): the first one, named “*Long Total Time Out Screen*” (A) is triggered if the user has been looking outside the screen for more than 10 seconds. The complementary one, “*Low Total Time Out Screen*” (A') is triggered when the user has looked outside the screen for less than 4 seconds. The amount of time the user has been looking outside the screen is computed in a cumulative way, i.e. by considering all the times s/he has looked outside the screen in the last three minutes. The first rule is triggered after the user observes the screen again.

If a user looks outside the screen for a long time, it may mean that s/he is inattentive, and for this reason we have opted for adding, as an action, a simplified and schematic version of the whole lesson and for auto-scrolling to the beginning of the added content. When the user has been looking at the screen continuously, s/he is instead likely to be reading the text and the adaptation action hides the simplified version of the lesson.

The second pair of rules, named “*Lost Attention*” (B) and “*Gained Attention*” (B'), consider the attention level parameter detected by the biosensor. The first one is composed of two sequential events: “neutral attention”, verified when the attention level parameter is between 60 and 100 (in a scale from 0 to 100, consistently with the Neurosky device eSense Meters) and “low attention” that is verified if the attention level is less than 20. Such thresholds have been chosen empirically to obtain a sensitivity sufficient but without high frequency of adaptations.

⁵ Moodle.org

⁶ <http://www.capterra.com/learning-management-system-software/>

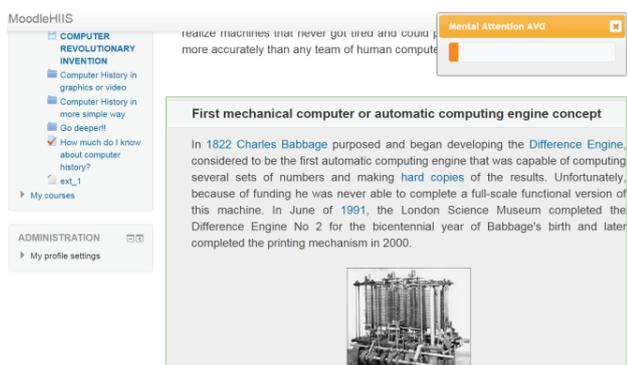


Figure 3: Standard (Not Adapted) version of the Moodle Course

When the loss attention rule is triggered (as in the case in Figure 3) then the adaptation completes the current observed paragraph by adding its simplified version, with a table summarizing the main events and concepts of the course (Figure 4 shows an example). The user attention is then stimulated and facilitates the attention increment.

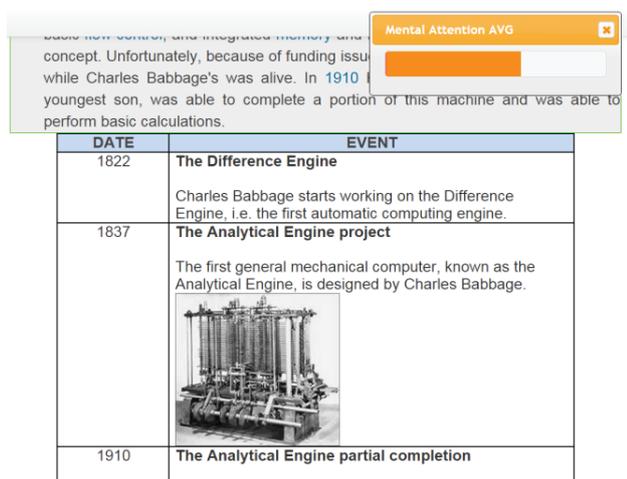


Figure 4: The adapted application showing the simplified contents.

The complementary rule, “Gained Attention”, is triggered when the average attention level changes from low to an high state. The actions adapt the page to the original version by hiding the simplified contents.

The last pair of rules, “Long Time Element High Blink” (C) and “Low Blink” (C’), considers the blinking parameter and the time the user has observed a paragraph continuously. The first rule is triggered when the same paragraph is observed for a long period and at the same time the user blinks often. The second is triggered when the number of blinks in the last three minutes is low. We made the assumption that watching the same section for a long interval and blinking often can indicate some trouble with the text. We thus provide the simplified version of the currently observed paragraph. The opposite occurs when

the blink rates drops, as this may indicate a more acceptable cognitive load.

USER TEST

The user test aimed to gather some empirical feedback on the proposed solution for adaptive Web UIs driven by physiological data for e-learning.

At the beginning the users received a textual description about the system functionalities, the devices used and the aims of the user test.

An eye tracker calibration was run once for each user, and consisted of watching a coloured circle moving across the screen for sixteen times. Users were presented with a Moodle course document about important achievements in the computer science field. Users had to memorise the main dates, inventions and inventors’ names. While reading, when the user’s physiological parameters changed and satisfied some rule conditions, a modal dialog appeared to inform the user about the detected event conditions and that an adaptation was being triggered. In that case users could accept or reject the adaptation.

After reading, users had to do a small exercise consisting of answering to 10 questions on an A4 paper sheet related to the contents aiming to check their actual learning. Finally, they filled in an on-line questionnaire providing personal data and assessments on the solution.

Seven subjects (5 male, 2 female) took part in the study, with age ranging between 26 and 37 (mean 31). All of them were workers at our institution, but were not involved in the project and did not receive any compensation for their participation. Regarding their education, four held a Master and three a Bachelor Degree. Only two of them had previous experience with LMSs (both with Moodle).

We observed users interacting with the system, during both reading the contents and answering related questions, and recorded session duration and any relevant event, i.e. how many times each kind of adaptation was proposed, and whether the user accepted or rejected it.

Reading the online course contents took between 8 and 16 minutes (mean 11). The number of adaptation events that occurred for each user varied between 10 and 20 (mean 14). Across all users, the most frequent event was C, that occurred 61 times. C’ was the least frequent, as it occurred only twice and only to one user (i.e. 6 users never raised this event).

In the questionnaire, users had to rate two aspects in a 1-5 scale, with 1 as most negative and 5 as most positive score. One aspect was the appropriateness of the rule to improve the contents comprehensibility (5: Very appropriate, 4: Appropriate, 3: Not particularly appropriate, 2: Not appropriate, 1: I was not aware of the adaptation). The other was the adaptation transition. (5: The transition was very good, 4: The transition was good, 3: The transition was

neutral, 2: The transition was bad, 1: The transition was very bad). For the appropriateness to improve comprehensibility, on average, (A/A') was rated 3.4, (B/B') and (C/C') were rated 3.7. The users seemed to prefer the group of events that occurred more often: A/A', which occurred 15 times and was rated 3.4; B/B' occurred 20 times and was rated 3.7; C/C' occurred 63 times and was rated 3.7

For the adaptation transition, on average, A/A' and (B/B') were rated 3.5, and C/C' was rated 3.1. For this aspect, which relates to the way adaptation is performed (e.g. insertion of additional content and automatic scroll to the newly added table), the trend is the opposite, as the C' got the lowest mean rating. Due to the small sample, we cannot draw statistical significance.

More information has been obtained from the logged events data. C occurred a total of 29 times on reading and 32 on answering. In detail, it occurred between 4 and 12 times to each user (mean 9). By considering that this event occurred quite frequently for some users, we argue that the condition threshold for the adaptation rule can be adjusted, e.g. by increasing the amount of blinks, which was set to 10 in this test. One user who noticed the C event frequency, declared that she often blinks when she is tired. This suggests that conditions on physiological parameters should not only be tailored to single (or a class of) individual(s), but should also take into account the current state of an individual. Given the unpredictability of some aspects (e.g. the blink frequency may depend on stress level), in future releases of the system we plan to let the user adjust conditions or temporarily deactivate rules.

C' is considered to be the complementary event of C, as it restores the original UI contents when the user blinks less than 4 times in three minutes. However, as already mentioned, it happened only twice over all the sessions (perhaps because of a too low threshold). It is worth pointing out that the actions triggered by this event were the same as the B' and A', thus users that never raised the "Low blink" could get back to the standard contents by focusing on the screen and/or re-gaining attention.

The following criticisms were provided: addition of content can be obtrusive and confusing; auto scroll to the added table causes loss of the reading point; eye gaze pattern analysis could be more meaningful than the amount of eye blinks. Suggestions were also collected: in case of attention loss, play a short sound, change page colour or emphasize key parts in the text; put additional content into a popup instead of the main page, and avoid auto scroll.

A final open question regarded the ability of the system to facilitate learning. Two participants answered positively; three said yes if adaptation rules show only the observed paragraph, and hide the others, when the user is particularly

focused, possibly reconsidering the user model parameters, and if the users were better informed of what adaptation is taking place. One answered negatively.

CONCLUSIONS AND FUTURE WORK

We have presented a novel approach to adaptivity in e-learning system based on a general architecture able to gather physiological data and adapt the application dynamically in order to better improve the user learning experience.

Future work will be dedicated to its application to mobile learning scenarios, which have not been considered in this work because the eye tracker considered is for desktop systems. We also would like to provide users with the possibility of customizing the adaptation rules, and content developers with an authoring tool that facilitates the creation of adaptation rules even for people without technical background. In addition, while the user test yielded positive feedback and was encouraging, we are aware that we need further empirical validation in order to extend and refine the set of adaptation rules that support dynamic improved learning.

REFERENCES

1. Akiki, P.A., Bandara, A.K., Yu, Y. RBUIS: simplifying enterprise application user interfaces through engineering role-based adaptive behavior. *Proc. EICS 2013*, ACM Press, 3-12.
2. Brusilovsky, P., Peylo, C. Adaptive and Intelligent Web-based Educational Systems. I. J. Artificial Intelligence in Education 13(2-4): 159-172 (2003).
3. Despotović-Zrakić, M., Marković, A., Bogdanović, Z., Barać, D., & Krčo, S. Providing Adaptivity in Moodle LMS Courses. *Educational Technology & Society*, 15 (1), 326–338.
4. Ghiani, G., Manca, M., Paternò, F., Porta, C. Beyond Responsive Design: Context-Dependent Multimodal Augmentation of Web Applications. *Proc. MobiWIS 2014*, LNCS Volume 8640, Springer Verlag, 71-85.
5. Magnaudet, M., Chatty, S. What should adaptivity mean to interactive software programmers? *Proc. EICS 2014*, ACM Press, 13-22.
6. Marchesi, M., Riccò, B. BRAVO: A BRAin Virtual Operator For Education Exploiting Brain-Computer Interfaces. *Proc. CHI 2013*, ACM, 3091–3094.
7. Nebeling, M, Speicher, M., Norrie, M.C. CrowdAdapt: enabling crowdsourced web page adaptation for individual viewing conditions and preferences. *Proc. EICS 2013*, ACM Press, 23-32.
8. Popescu, E., Badica, C., Moraret, L. Accommodating Learning Styles in an Adaptive Educational System. *Informatica (Slovenia)* 34(4): 451-462 (2010).