XML-based learning scenario representation and presentation in the adaptive e-Learning environment

P. Kazienko, J. Sobecki

Department of Information Systems, Wroclaw University of Technology
Wyb. Wyspianskiego 23, 50-370 Wroclaw, Poland
{kazienko,sobecki}@pwr.wroc.pl

Abstract: E-Learning environments nowadays should adapt to their users and computer platforms they are using. In this paper the learning scenario collaborative recommendation based on consensus method using XML is presented. We present the XML description of the learning scenario and its components as well as the link database and XSL transformations for presenting them on given platforms in the way suitable for particular students.

1 Introduction

The Information Society development requires that their members are able to utilize new ways of acquiring knowledge and new skills. Traditional learning environments are necessary for primary and secondary education, however for higher, postgraduate and other forms of additional education, different modern forms of education and training, such as e-Learning or Computer Based Teaching, are becoming vital nowadays. This is mainly because the times when knowledge acquired by people in the traditional education institutions such as schools or universities were sufficient for the whole life, have gone for ever.

Nowadays e-Learning environments based on the web technologies are attracting increasing number of very differentiated users. There are many reasons for such environments (especially Distant Learning) being so popular [11]: facilitating different retraining programs for participants from different areas; providing access to specialist centers; providing equal opportunities in any aspect; promotion prices for non-campus students; making courses more student directed; revitalizing less popular courses; linking exploded campuses; improving the quality of courses.

E-Learning environments have also some disadvantages, from which the almost complete reduction of personal contacts between students and teachers is of the greatest importance. The lack of these contacts hinder appropriate understanding between students and teachers what results in different problems in the education process, for which the problems of Tutoring Strategy (TS) selection are ones of the greatest significance.

The TS is defined as a combination of means and methods used in the whole didactic process to increase its effectiveness. The TS may be realized by the Learning Scenario (LS), which is the selection of the knowledge pieces as well as their sequence, connections and form of presentation. In many e-Learning environments despite the differences among students and their progresses only single TS is applied. In works [9, 14] the solution of this problem by means of application of collaborative LS selection, which offers adaptation of the LS according to other students' experiences, was presented together with its application. In this paper we presented the application of XML to represent most of the system elements, such as LS instances and student model but also to display the education material on the particular platforms. The XML proved to be suitable tool for representation and presentation of very complex data also in adaptive interface recommendation [15] and adaptive tutoring strategy selection [14] systems.

In the following section the XML-based strategy representation together with its components is presented. In the third section the XML-based representation of student model and the process of LS adaptation are given. The following section describes link extraction and utilization in learning scenario presentation. In the fifth section the method for the actual scenario extraction and presentation on the given platform are shown. The conclusions and future works perspectives are presented in the summary.

2 XML-based learning scenario representation

The different TS are usually designed to fit differences in students' learning styles, which in turn are consequences of differences in cognitive styles. According to the work [2] for each learning perspective one of the two contradictory values could be assigned: perception (sensory or intuitive), input (visual or auditory), organization (inductive or deductive), processing (active or reflective) and understanding (sequential or global). For each learning perspective corresponding teaching style could be assigned accordingly: content (concrete or abstract), presentation (visual or verbal), organization (inductive or deductive), student participation (active or passive) and perspective (sequential or global). Tutoring strategies are characterized with any combination of values assigned to the above mentioned perspectives. We could obtain a numerous set of different strategies if we want to meet all of them.

In the implementation of e-Learning environment for Polish traffic regulation (presented in [14]) only four main types of strategies were distinguished: textual, graphical, animated and active. It was obviously a kind of simplification of the above presented model. The actual tutoring strategy was built from elements of four different types, however only specified combinations of elements were supposed to be valid.

It is possible to design more flexible environments that are able to be used by very differentiated population of learners who are using various computer platforms. Let us consider the tutoring system which role is to present the material of some courses. We can divide the whole material into some parts, which could be called *concepts* that in turn are to be learned by a student. Depending on the learning strategy the sequence of concepts may be different. Also the concepts themselves could be presented in many different ways. For many traditional web-based e-Learning environments the concepts are presented by a sequence of hypermedia pages [8] - see fig.1. But when we consider not only differences among learners but also among computer platforms they are using, the model should not perceive the pages (for example HTML) as atoms.

In the proposed model we should also define the pages as separate elements, but on the pages we should distinguish other chunks of information that could be presented to the user, according to the user's and platform's requirements. The information presented on the pages could be in the following forms (media): text, image, sound, video or animation, and interactions (for example implemented in Macromedia Flash). The page definition contains also information about the content, its sequence of appearance and links to other pages.

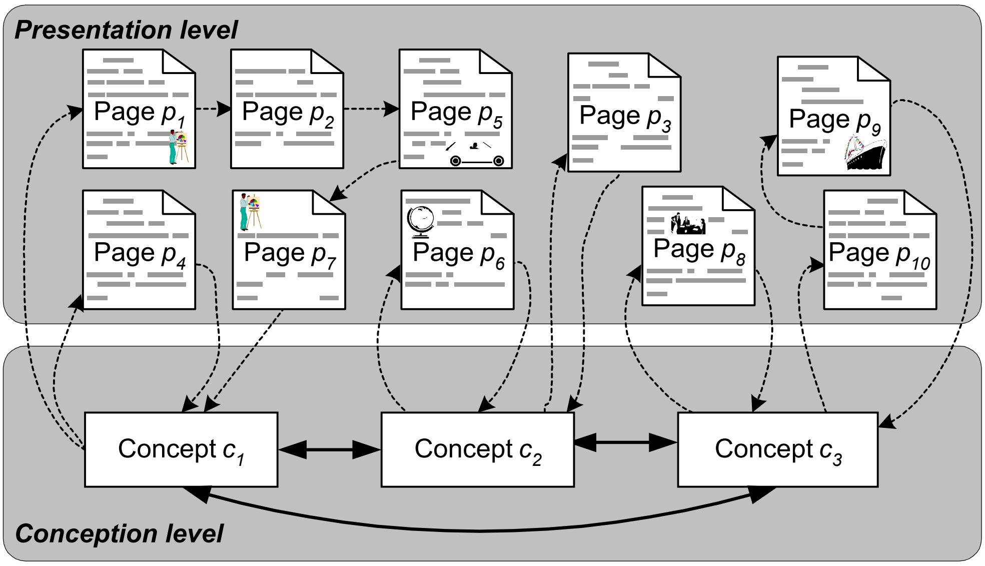

Fig. 1. Knowledge structure and its presentation.

The complete learning strategy contains not only concepts but also tests that verify the knowledge acquired by the student *st* belonging to the set of students *St* ($st \in ST$). The tests are also to control the learning process and help to decide whether repetitions or even changes in the learning strategy are necessary. So the learning scenario could be defined in the following way. Let *C* be the finite set of concepts to be learned and *T* be the finite set of tests checking the knowledge of all the concepts from the set *C*. The learning scenario s_{st} for a given student *st* (belonging to the set of scenarios $s_{st} \in S$) is defined as any reasonable order in *C*, where for each concept $c \in C$ there is assigned corresponding test $t_c \in T$. This, however, defines only *the conception level* of the learning scenario. To define *the content level* we must introduce the notion of pages, which are the content containers. Let *P* denote the set of pages, where each $p \in P$ contains corresponding resources in form of various media: texts, images, tables, audio, animation, video, interactive elements and hypermedia links. The content of each $c \in C$ is defined as reasonable ordered subset of pages $O \subset P$. All the resources have also information assigned, which define the computer platforms (*F*) they could be displayed or played on.

The XML representation of the learning scenarios could be defined as follows. First we shall define the XML document containing all the information about the atom elements (see the fragment of the XML file `media_el.xml` below):

```
<content_element element_id=”e0021” type=”video” storage=”/video/001.avi”>
     <platforms>
            <platform platform_id=”p01”>pc</platform>
            <platform platform_id=”p04”>kiosk</platform>
     </platforms>
</content_element>
<content_element element_id=”e0274” type=”text” storage=”/texts/010.txt”>
     <platforms>
            <platform platform_id=”p01”>pc</platform>
            <platform platform_id=”p04”>kiosk</platform>
            <platform platform_id=”p03”>pda</platform>
            <platform platform_id=”p02”>mobile</platform>
     </platforms>
</content_element>
```

Then we are able to define the sample page (see bellow). The concepts and the learning scenarios could be defined respectively:

```
<page page_id=”p121”>
     <element element_id=”e0102”/>
     <element element_id=”e0133”/>
     <element element_id=”e2109”/>
</page>
```

3. Learning scenario adaptation

In works [9,14] the method for consensus based learning scenario adaptation is presented. It is based generally on the ideas of collaborative recommendation [12]. The overall adaptation architecture is presented in fig. 2. The adaptation is based on the consensus methods, which are seeking for such solution that is the most representative for some subset of the whole set of elements, called also the profile [13].

We assume that in the e-Learning environment the population of students are learning one particular course. For each student the appropriate learning scenario in recommended. The recommendation in the first phase of the system life is based only on the expert knowledge, however with the growth of the number of learners the system is collecting its experiences that are the basis of the consensus based learning scenario adaptation. The recommendation of the scenario is made upon the consensus of the successful learning scenarios of the group of similar learners. To determine the groups, we introduce the notion of the learner profile and also the distance function between pairs of the learner profiles. Then using for example popular k-means clustering algorithm the learners are grouped by their profiles.

According to [7] the user model in the adaptive system consist of the user data and the usage data. The user data characterizes the user of the system and contains his or her demographic data (name, sex, address, occupation, education, customer data, psychographic data, etc.), user knowledge, user skills and user interests. The usage data contains information that may be observed directly from the user’s interaction with web-based system [6]. In the e-Learning environment the user data may also contain information about learner cognitive characteristic [2] and the usage data contains the learners' history of learning scenarios already passed together with information of corresponding test results.

The overall architecture for the learning scenario adaptation is presented in fig. 2. The result of the adaptation is the learning scenario recommendation for a given student. The scenario contains the collection of pages with corresponding test.

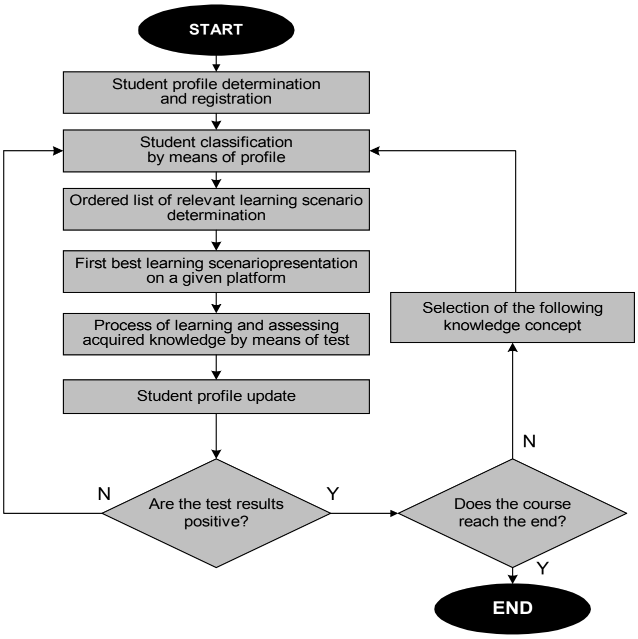
START
Student profile determination and registration
Student classification by means of profile
Ordered list of relevant learning scenario determination
First best learning scenariopresentation on a given platform
Process of learning and assessing acquired knowledge by means of test
Student profile update
Selection of the following knowledge concept
N
N
Are the test results positive?
Y
Does the course reach the end?
Y
END

Fig. 2. Architecture for consensus-based learning scenario adaptation.

4. Navigational link extraction for learning strategy

The best nowadays solution for hyperlink modelling seems to be the XML Linking Language (XLink) [1,3,4]. There are two main link types provided by XLink standard: simple (similar to HTML anchors) and extended. The second ones are allowed to be stored outside the source document what is very useful for adaptive e-Learning content generation. A single `xlink` contains two general XML child elements: `locators` and `arcs`. Locators store references - bindings (URIs) to linked resources and assign them labels and titles. Titles can be later presented to a user as target resource names (as the content of `<a>` element in HTML). In e-Learning systems locators are used for pointing at both scenario media and particular presentation pages. Locators may indicate any element within an XML external document using XPath expression (XPath is the standard retrieval language for XML). In the example shown below these expressions have only the form of identifier references (e.g. #e0015, #p121) that means pointers to elements containing any attribute of ID type with certain value (e0015 and p121 respectively):

```
<link_database xmlns:xlink="http://www.w3.org/1999/xlink"
               xlink:type="extended">
 <name xlink:type="title">Link for scenario</name>
            <!-- resource mapping (binding and indicating) -->
 <media xlink:type="locator"           xlink:title="Video no. 1"
        xlink:href="media_el.xml#e0021" xlink:label="IntroVideo" />
 <media xlink:type="locator"           xlink:title="Video 2"
        xlink:href="media_el.xml#e0015" xlink:label="IntroVideo" />
 <media xlink:type="locator"           xlink:title="Video no. 5"
        xlink:href="media_el.xml#e0024" xlink:label="BasicCourseVideo"/>
              ...
 <page  xlink:type="locator"           xlink:title="Page no. 1"
        xlink:href="pages.xml#p121"     xlink:label="Page1"/>
 <page  xlink:type="locator"           xlink:title="Page no. 2"
        xlink:href="pages.xml#p142"     xlink:label="Page2"/>
              ...
           <!-- arcs: resource connections -->
 <video xlink:type="arc"              xlink:title='Video Basic'
        xlink:from="IntroVideo"       xlink:to="BasicCourseVideo" />
 <video xlink:type="arc"              xlink:title='Video Intro'
        xlink:from="BasicCourseVideo"  xlink:to="IntroVideo" />
 <pages xlink:type="arc"              xlink:title="Pages' links"
        xlink:from="Page1"            xlink:to="Page2"
              ...
</link_database>
```

Arcs create connection between all locators with specified label. An arc begins in any resource with the label equal to arc's from attribute value and ends in a resource pointed by to attribute. Please note that a single arc can bind many resources simultaneously, because many resources are allowed to possess the same label. For example, the first arc (Video Basic) in the link database presented above makes at least two connections: from Video no. 1 to Video no. 5 and from Video no. 2 to Video no. 5. The third arc (Pages' links) links from Page 1 to Page 2.

Hyperlinks starting on a source page presented to a user may be merged dynamically with the page content using of the external link database similar to described above and appropriate XSL transformation. We only need to retrieve suitable resources pointed by arcs (fig.3). For particular source page its identifier and identifiers of media resources (content elements) presented on it are known after adaptive scenario selection. Having these identifiers, labels of suitable locators can be retrieved from the link database. All arcs with attribute from equal to obtained locator labels point at target resource owing to their to attribute. Links of required form (suitable for user platform) can be generated from addresses of these target resources. The link generation, resources and arcs retrieval are done within proper *templates* in an XSLT stylesheet. These templates are then incorporated into the final XSLT so obtained links are outbound [1].

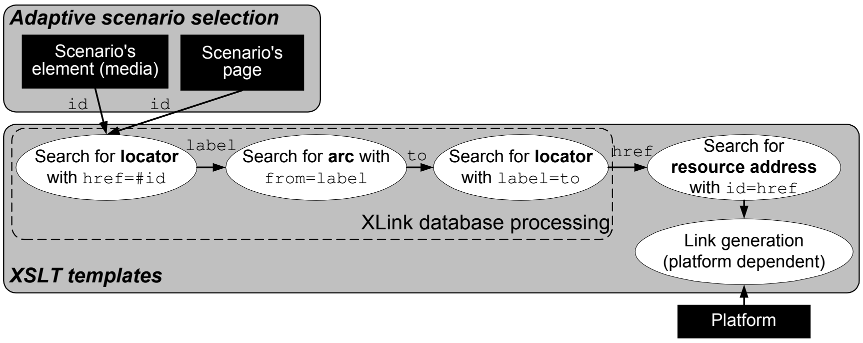
Adaptive scenario selection
Scenario's element (media)
Scenario's page
id
id
Search for locator with href=#id
label
Search for arc with from=label
to
Search for locator with label=to
href
Search for resource address with id=href
XLink database processing
Link generation (platform dependent)
XSLT templates
Platform

Fig.3. The concept of link generation form XLink database using XSLT

5. Learning strategy content presentation on different platforms

A content presented to a student *st* depends on the three issues: layout (*L*), presentation platform (*F*) and learning scenario determined adaptively (s_{st}) in form of the sequence of pages with information content and tests.

The layout reflects esthetic preferences specified directly by a learner that are stored in the e-Learning system. It can be realized during the registration process: the system presents snapshots of all possible layouts to a learner, who chooses the closest one (the best looking for him or her). These preferences may also be changed before the presentation starts. The set of layouts is fixed and dependent on the e-Learning environment. Thus, layout selection is based on manually expressed preferences. However it may also be done automatically by analyzing previous user's e-Learning sessions - the idea similar to link recommendation from [6].

The second factor is a platform recognized by the system. This limits presentation possibilities of the particular scenario media types - not all are able to be shown on a given platform. These platforms are defined not only by the hardware (PDA, mobile phone, computer-host, printer) but also by the operating system (Windows, Linux) and even the presentation tool, e.g. browser (IE 6.0, Netscape 7.1). The platform has the biggest influence onto the output format of presentation: HTML (for specific browsers), HTML Basic and WML (for mobile phones), PDF (for printers).

Both the platform and layout determine main transformation process (fig. 4). The system generates (or retrieves from static repository) the independent stylesheet (XSLT) for each pair layout-platform (L_i, F_j). This stylesheet, including so called *templates*, describes the way of data transformation from the source format into the format, which is appropriate for the specific platform. The layout *L* is responsible only for the location of particular scenario elements on the output page. The transformation language (XSLT) enables to obtain almost any textual format for target documents, including HTML, WML, TXT, XSL FO (XSL Formatting Objects), etc. [5]. From XSL FO documents it is also possible to generate PDF or Postscript files, which are suitable for printers, using free software, e.g. FOP, quite easily.

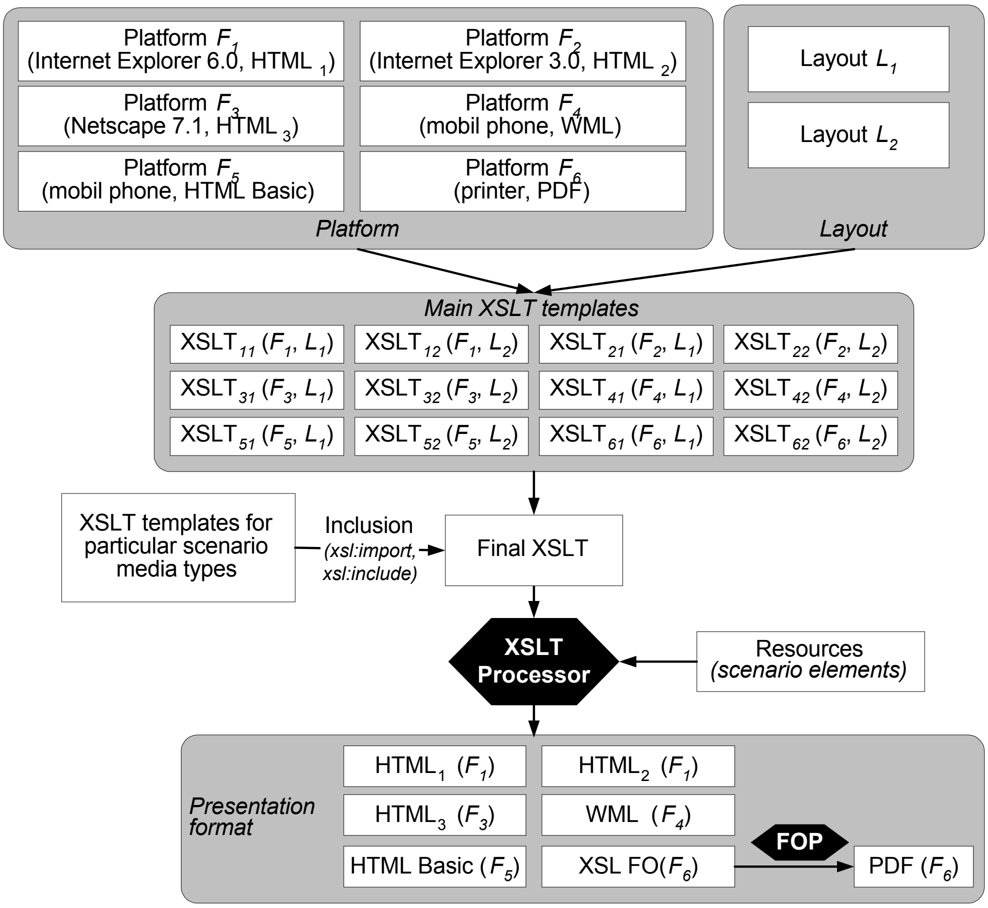
Platform F_1 (Internet Explorer 6.0, HTML $_1$)
Platform F_2 (Internet Explorer 3.0, HTML $_2$)
Platform F_3 (Netscape 7.1, HTML $_3$)
Platform F_4 (mobil phone, WML)
Platform F_5 (mobil phone, HTML Basic)
Platform F_6 (printer, PDF)
Platform
Layout L_1
Layout L_2
Layout
Main XSLT templates
XSLT $_{11}$ (F_1, L_1)
XSLT $_{12}$ (F_1, L_2)
XSLT $_{21}$ (F_2, L_1)
XSLT $_{22}$ (F_2, L_2)
XSLT $_{31}$ (F_3, L_1)
XSLT $_{32}$ (F_3, L_2)
XSLT $_{41}$ (F_4, L_1)
XSLT $_{42}$ (F_4, L_2)
XSLT $_{51}$ (F_5, L_1)
XSLT $_{52}$ (F_5, L_2)
XSLT $_{61}$ (F_6, L_1)
XSLT $_{62}$ (F_6, L_2)
XSLT templates for particular scenario media types
Inclusion (xsl:import, xsl:include)
Final XSLT
XSLT Processor
Resources (scenario elements)
Presentation format
HTML$_1$ (F_1)
HTML$_2$ (F_1)
HTML$_3$ (F_3)
WML (F_4)
HTML Basic (F_5)
XSL FO(F_6)
FOP
PDF (F_6)

Fig.4. XSL Transformation for platform and layout

The pair (L_i, F_j) corresponds only to the main stylesheet templates but the page itself includes data specific for particular scenario elements. For each element (media) type exist in the system template set (XSLT) responsible for transformation of this type into output format (fig. 5). Please note that element templates (like link generating templates) are also platform dependent. All these necessary templates are attached to the main stylesheet using XSLT import mechanisms (`xsl:include` or `xsl:import`). Main XSLT stylesheet extended with element and link templates is applied to the resources (especially XML documents) related to given scenario page. The page in platform dependent format is obtained as the output of transformation process.

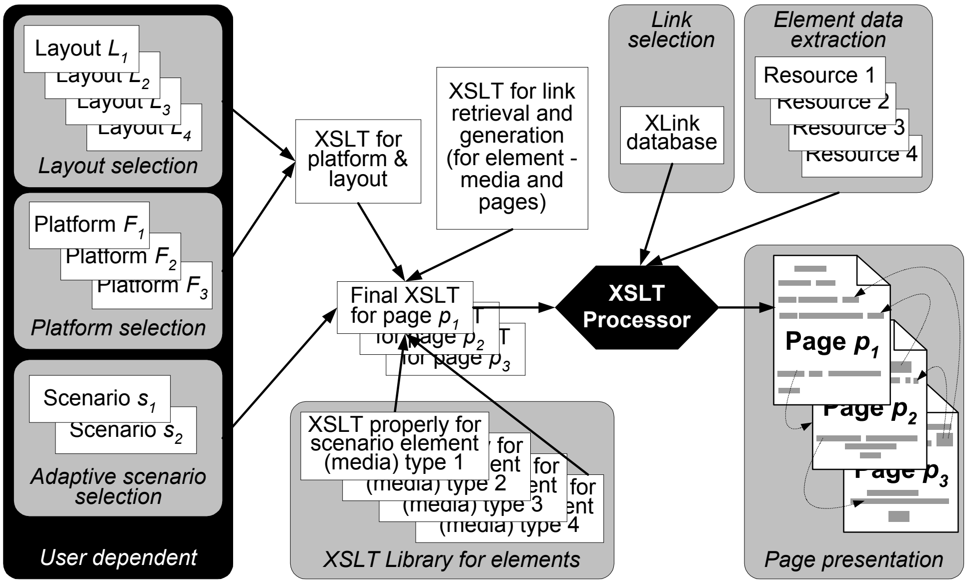

Fig.5. Content presentation process

6. Conclusion

In the paper the XML-based implementation of adaptive learning scenario is presented. The adaptation of the learning scenario is based on the consensus methods, which recommends for each student an ordered list of learning scenarios. They are ordered with decreasing relevance to the given learner. As some of the media resources, which are placed on pages that are elements of learning scenarios, could not be displayed on some computer platforms, the given scenario is rejected and the next one is considered.

The presented adaptive e-Learning environment architecture is rather simplified so in the future works we shall consider other more complex approaches. We can for example accept such scenarios which contain elements that could not be displayed on the given platform but such media should be replaced by other acceptable resources. This however may cause the decrease of the efficiency of the given scenario so we shall also consider whether the second best scenario is not better than the current one with modifications. The other approach is to consider the system platform in the student profile and to classify them respectively or consider direct interface adaptation [15].

References

1. DeRose S, Maler E, Orchard D (eds) (2001) XML Linking Language (XLink) Version 1.0. W3C Recommendation, World Wide Web Consortium, 27 June 2001, http://www.w3.org/TR/xlink/.
2. Felder R, Soloman B (1998) Index of Learning styles http://www2.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/ILS-a.htm.
3. Gwiazda K, Kazienko P (2001) XLink - the future of document linking. Proc. of the Information Systems Architecture and Technology ISAT 2001. Wroclaw University of Technology: 132-139 http://www.zsi.pwr.wroc.pl/pracownicy/kazienko/pub/ISAT2001/XLink.ps.
4. Kazienko P (2002) Finesse of hyperlinks. Tele.net Forum 9: 24-29. (in Polish).
5. Kazienko P, Gwiazda K (2002) Taking XML seriously. Helion, Gliwice (book in Polish).
6. Kazienko P, Kiewra M (2003) Link Recommendation Method Based on Web Content and Usage Mining. New Trends in Intelligent Information Processing and Web Mining Proceedings of the International IIS: IIPWM´03 Conference, Zakopane, June 2-5, 2003, Advances in Soft Computing, Springer Verlag: 529-534 http://www.zsi.pwr.wroc.pl/pracownicy/kazienko/pub/IIS03/pkmk.pdf.
7. Kobsa A, Koenemann J, Pohl W (2001) Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships. Knowledge Eng. Rev. 16(2): 111-155.
8. Kukla E, Daniłowicz C, Nguyen NT (2003) Reconciling inconsistency in selection of optimal strategy in multimedia intelligent tutoring systems. Proc. of HERCMA 2003, Athens 24-27 Sep. (to be published).
9. Kukla E, Nguyen NT, Sobecki J, Danilowicz C, Lenar M (2003) A Model Conception for Learner Profile Construction and Determination of Optimal Scenario in Intelligent Learning System. Lecture Notes in Computer Science 2774: 1216-1222.
10. Kukla E, Nguyen NT, Sobecki J (2001) The consensus based tutoring strategy selection in CAL systems. World Transactions on Engineering and Technology Education 1: 44-49.
11. Lennon JA (1997) Hypermedia Systems and Applications – World Wide Web and Beyond. Berlin: Springer – Verlag.
12. Montaner M, Lopez B, de la Rosa JP (2003) A Taxonomy of Recommender Agents on the Internet. Artificial Intelligence Review 19: 285-330.
13. Nguyen, NT (2002) Consensus systems for conflict solving in distributed systems. Journal of Information Sciences 147: 91-122.
14. Sobecki J, Morel K, Bednarczuk T (2003) Web-based Intelligent Tutoring System with Strategy Selection Using Consensus Methods. New Trends in Intelligent Information Processing and Web Mining Proceedings of the International IIS: IIPWM´03 Conference, Zakopane, June 2-5, 2003, Advances in Soft Computing, Springer Verlag: 105-109.
15. Sobecki J (2003) XML-based Interface Model for Adaptive Web-based Systems User Interfaces. Proceedings of International Conference on Computing Science 2003. Lecture Notes in Computer Science (Springer-Verlag) 2660: 592-598.