Rapporto n°22 2004

Adaptable Web content for e-learning communities

Alfonso Quarati

[1] INTRODUCTION

The cost of design and creation of educational material to be used in the context of web-based courses is expensive and time consuming. Other than the usual costs associated with the production of any kind of web content a particular effort is required to design pedagogically-driven content [1]. In fact, in such cases the author must bear in mind, not only the quality of the content with respect to the specific cognitive domain, but also its adequacy w.r.t. students' educational requirements (e.g. goals, past experiences, learning preferences, …)[2]. However educational institutions are often characterized by the production of very similar educational resources by educators who carry out teaching activities in strictly related fields. In order to foster relevant savings, the policy of the sharing of materials is often pursued both at an intra or extra organizational level [3].

To meet the above requirements we suggest integrating two different methodologies: 1) to reuse existing materials and 2) to realize Web content which tailors to different user needs. As to 1) several efforts have been carried out, in the past decade, by worldwide consortiums and organizations such as IEEE Learning Technology Standards Committee [4], the IMS Consortium [5], ADL SCORM [6], towards the definition of standards for the production and the exchange of educational resources based on meta-data specification.

As to 2) a personalizable system is an application that adapts its behaviour to the goals, interests, and tasks of individual users or groups of users [7]. Several personalizable systems in different application areas namely, e-commerce [8], narrative [9], mobile computing [10], information retrieval [11], education [12], have been developed in the past decade. Very often, the main emphasis of these works is to point out the way they adjust to the content to users' needs, stressing the specific tailoring features of the system. Though agreeing on this aspect, we believe that another point should be highlighted when designing a personalizable system. The capability to overturn the "one-size-fits-all'' approach typical of most web based systems, by creating user-aware materials once, but fitting different user needs, enforces the possible re-use of the content.

In this paper we explore an easy-to-use methodology aimed to optimise the design, construction and maintenance processes of Web-based educational material, which leverages on adaptive features to foster re-use and sharing between educational communities. Our approach is easier than those proposed by the cited organizations in the sense that it borrows the principal idea of describing learning material through meta-information (about properties and structures of the educational contents and the existing relationships between them), but discards the inherent complexity typical of their richer categorization schemas that may overwhelm the authors' task. This simplification will favour, in our opinion, a more light-weight and more rapid production and delivery of educational content.

The paper is organized as follows. Section two discusses the modelling process on which our methodology is based. Section three presents the rationale behind the implementation choices. Section four details the personalizable techniques adopted to realize the system. Finally, section five presents some concluding remarks.

[2] A 4-LAYERED MODEL FOR REUSABLE AND ADAPTABLE COURSEWARE

There is a common understanding that the design of a personalizable hypermedia system relies on the coexistence and the interaction of different modelling layers [13]. Generally, one of those layers models the content domain, another models the user and his interaction with the system on the basis of an adaptation domain. We find it useful to distinguish a further layer, that we call the *delivery model*, which describes the way in which the adaptation engine, i.e. the system component which supervises the tailoring process, has to adapt the content to the user according to the author's pedagogical choices.

2.1. Content model

The content domain is modelled by a set of *objects,* each one representing a concept according to the author's view and objectives. Objects are defined starting from a collection of *subcomponents*. Each of these subcomponents may represent some elementary meaningful information which can be expressed in the form of text, image, audio/video, or link to some other Web resource with which the object is in some associative relationship. Each subcomponent is represented by an identifier, we further see that it is indeed an URI [14], which allow the author to create content in a modular and hence re-usable way: the same subcomponent can be selected by different authors to build a variety of objects to be used in different *contexts*. Fig. 1 schematises the tree-like structure of two objects O1 and O2 with the shared subcomponent bc4 highlighted. Furthermore, when a URI refers to some external document it will be possible for an author to incorporate third parties' resources to realize a more flexible and open informational environment as has been suggested by other authors [15].

Moreover, to reinforce the exchange of content between learning communities, each subcomponent is divided into *local fragments*, each one corresponding to a destination language. In this way the same content may be represented, and hence understood by students with diverse mother tongues. Even if some topics can be thought not free from language or cultural constraints (e.g. history, geography, literature which are generally bound to the country and hence to the language), many others are. For example, a tutorial in object-oriented programming, a monographic course about Socrates, a series of lessons about First Order logic, are largely independent from language-country peculiarity. So it is reasonable to consider learning scenarios in which content is written in a language, and then translated into others.

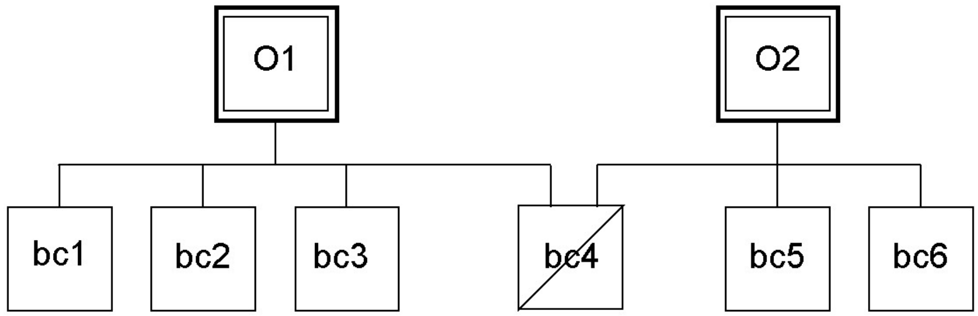


Fig1. Graphic representation of the hierarchic structure of objects O1 and O2.

The overall structure of the hypermedia is then established by the author who, each time, chooses and hierarchically organises a collection of these objects, also called, for short, a forest. In this way the author is able to describe different perspectives of the content domain according to his own expositive needs and to the expected final user.

Note that so far, the peculiarity of the destinations of use of the content, i.e. e-learning environments, are not relevant to this modelling step so that we can generally speak of 'content' without any specification. Further on, we specify that a forest, in a learning scenario, is indeed a collection of hypermedia educational material in the form of an electronic textbook, which instantiates a course, or a class lecture.

2.2. Adaptation model

A relevant aspect to take into account for an author developing personalizable Web content and specifically educational one, is the inherent conceptual difficulty of each concept compared with that of the others occurring in the same context. For example, in an analysis course, the concept of integral is usually considered more difficult than the concept of limit, that often precedes the former. Fixed the context of use, it is meaningful to associate to each object O a weight $\omega(O)$, which represents the cognitive contribution of that object within that context. The same can be applied to object level, associating a weight to each of its subcomponents. We call *absolute weight* of an object O, $|O|$, the maximum weight between its subcomponents.

Similarly it is also useful to identify for a textbook T, that is a hierarchy of objects, two parameters: $\Omega(T)$ the *weight* of T, is defined as $\max(\{|O|+\omega(O), \forall O \in T\}$ while $|T|$, the *absolute weight* of T, is the maximum of the weights of the objects of T. By definitions we always have $|T| \leq \Omega(T)$. For the textbooks A e B in Fig. 2 we respectively have |A|=1 Ω(A)=2 (with $\omega(O3)=0$ and $|O3|=2$), and |B|=1 Ω(B)=3 ($\omega(O3)=1$ and $|O3|=2$). Intuitively one can see the weight $k=\Omega(T)$ of a textbook T as representative of the k successive levels of reading available to students.

Based on the assignment of cognitive weights to concepts, and to the characterization of student knowledge, as explained in next section, the system, specifically the adaptation engine component, is able to finely tailor the content to user actual needs. Following this approach, starting from a fixed content model, an author/teacher is able to construct a variety of views, also called *adaptation models*, each one corresponding to a different context of use, so realizing a multiplicity of tailored learning paths.

2.3. The user model

Students' goals and interaction with the system may differ depending upon their different backgrounds, interests, motivations, contexts of use, which result in different visited content or followed links. Moreover user behaviour varies with time according to his evolving perception of the knowledge acquired. A user model is required to represent such a student's characterization. A particular relevance while modelling the user is given to his knowledge of the concepts being learnt. Generally, a mapping between concepts and the level of user's knowledge about them is realized; this represents the "knowledge" of the user. This mapping is commonly known as an *overlay model*. Several adaptive hypermedia systems define their specific overlay model [16]. Continuing this train-of-thought our choice is to characterize the user by his *knowledge state* with respect to a course C, as the 3-tuple $<i, a, f>$ of integers, $0 \leq i \leq a \leq f \leq \Omega(C)$, where i represents the initial knowledge of the student, f is the target knowledge to be acquired by the student upon completion of the course and a is his *actual knowledge*. In this way at any moment it is possible to determine the concepts (i.e. their associated corresponding objects/subcomponents) which are *visitable* by a student. This is accomplished by comparing the actual knowledge of the student with the weighted trees and filtering out the subcomponents or objects which have a greater weight than the student's actual state. These are indeed concepts that he is not ready to learn.

The case of two students with the same actual knowledge of 0, while accessing two different (though similar) courses A and B is schematised in Fig. 2. The objects and the subcomponents (with weights equal to 0) that are visitable by the two students are highlighted in grey. Note that object O3 is visible to the student exploring course A while it is not accessible for a user consulting course B. This depends on the different authors' choice in weighting O3 in the two diverse contexts.

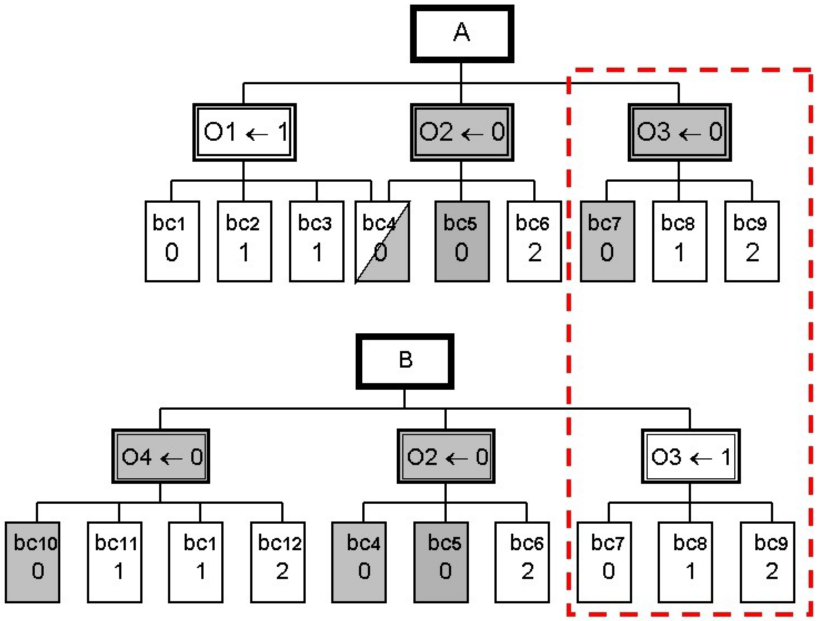


Fig. 2. The tree-like structure of two courses A and B. Note that shared subcomponent bc4 in course A, is accessed through object O2 not through O1. In an analysis course, for example, this is the case if bc4 stands for the "definition of continuous function", and is used in O2 which introduces the concept of "derivate of a function", and in O1 which informs about the concept of "integral over an interval" (a more advanced concept).

As to the assignment of value to the 'knowledge state 3-tuple', this is established by the teacher who generally selects the initial and the final states, and chooses the strategy to be followed by the system so to update the user's actual state according to the delivery model, as discussed in the following section.

To permit the adaptation engine to properly accomplish the personalization tasks, the user data are collected and recorded by the system. The knowledge state 3-tuple along with other student data, such as demographic information, reading style preferences, language, are maintained into the so-called *student registry*. The student behaviour, (e.g. accessed objects, tests solved), is tracked in *activity records* depending on the fixed delivery model.

2.4. The Delivery model

Different teaching contexts may influence the way in which the student progresses in acquiring new and more complex knowledge about a topic. For example, in a two-semester university course, students are only allowed to access the second-semester syllabus after having successfully passed a set of examinations based on concepts learnt during the first-semester. Instead, an employee following a training-on-the-job course by distance-learning, could be left free to 'jump' from easier to more difficult concepts and vice-versa according to his actual working needs.

To take those diverse requirements into account, we propose an approach which allows the educator each time to select the most suitable pedagogical strategy to be followed according to the teaching context. So, independently from the adaptation modelling of the concepts, and from the state of the student's actual knowledge, it is possible to deliver different 'contextually aware' learning experiences. The delivery model is characterized by the specification of sets of logical conditions, called *delivery criteria.* A delivery criterion determines which part of the content domain is available to an individual user, at a certain time. More precisely each delivery criterion is represented by a n-tuple of delivery conditions that have to be satisfied by a learner before going deeper into the content domain. These conditions assert the availability of the content according to the following.

Let $\Delta(T) \equiv \langle p_0, p_1, \ldots, p_{\Omega(T)} \rangle$ be a delivery criterion for a textbook T. Where $p_h \equiv \langle \delta_h, V \rangle$ such as:

- $V \equiv \langle v_0, v_1, \ldots, v_q \rangle$ is a subset of the objects of T with $\omega(v_i) \leq h$, $0 \leq i \leq q$.
- $\delta_h : V \rightarrow$ [true, false] is called *delivery function of level h*.

On this basis we said that a *delivery condition* $p_h \equiv \langle \delta_h, V \rangle$ has satisfied by a user *iff* for each $x \in V$, $\delta_h(x) = \text{true}$.

In other words, the satisfaction of condition p_j allows the student to augment his knowledge by accessing 'j+1' weighted objects.

To clarify the underlying idea, let us follow three examples of delivery criteria for a textbook T such as $\Omega(T) = 2$.

1. Let t_1, t_2, t_3, t_4 four "test page" objects of T such as $\omega(t_i) = 0$, $0 \leq i \leq 4$. Each t_i contains subcomponents (e.g. single-answer tests) having weights that comprise between 0 and 2.

   Let $\Delta_1(T) \equiv \langle p_0, p_1, p_2 \rangle$ with $p_h \equiv \langle \delta_h, V_h \rangle$, $0 \leq h \leq 2$ such as:
   - $V_h \equiv \langle t_1, t_2, t_3, t_4 \rangle$ and
   - δ_h be informally defined as:

$\delta_h(t_i)$: '*having all the subcomponents of* t_i *with weight* h *being correctly met?'*

2. Let x_1, x_2, x_3 three 'compulsory reading' objects of T such as $\omega(x_i) = 0$, $0 \le i \le 3$. Each x_i contains subcomponents having weights that comprise between 0 and 2.

   Let $\Delta_2(T) \equiv <p_0, p_1, p_2>$ with $p_h \equiv <\delta_h, V_h>$, $0 \le h \le 2$ such as:
   - $V_h \equiv <x_1, x_2, x_3>$ and
   - δ_h be informally defined as:

     $\delta_h(x_i)$: '*has page* x_i *been accessed?*'

   (Note that this last definition implicitly takes count of any new access required to discover novel and 'heavier' subcomponents of a page.)

3. With x_1, x_2, x_3 and t_1, t_2, t_3, t_4 as above, we can define a third criterion that mixes the previous :

   Let $\Delta_3(T) \equiv <p_0, p_1, p_2>$ with
   - $p_h \equiv <\delta_h, V_h>$, $0 \le h \le 1$ such as $V_0 = V_1 \equiv <x_1, x_2, x_3>$ and $\delta = \delta_0 = \delta_1$ be informally defined as:

     $\delta(x_i)$: '*has page* x_i *been accessed?*'
   - $p_2 \equiv <\delta_2, V_2>$, such as $V_2 \equiv <t_1, t_2, t_3, t_4>$ and δ_2 be informally defined as:

     $\delta_2(t_i)$: '*having all the subcomponents of* t_i *with weight* 2 *being correctly met?'*

As several criteria may be specified for a course, the teacher is free to choose the most appropriate one for the situation at hand. Based on the teacher choice the adaptation engine, once it has verified that all conditions for some fixed criterion have been satisfied by a student, updates his user model, thus allowing the student to discover new and more advanced content. As explained in more detail in the following section, the verification of a criterion is carried out through the examination of the student's activity records.

The introduction of the delivery model, in our opinion will further the re-use and the adaptability of a course to different learning contexts and individuals' needs, and more generally the sharing of educational material between educators.

[3] SYSTEM DESCRIPTION

A well accepted classification of personalizable systems subdivides them into two main (often overlapping) categories depending on the way the tailoring process is initiated, namely by *adaptable* and *adaptive* systems [17]. In the former the user explicitly exercises some choice over the kind of adaptation he prefers. In the latter the system adapts autonomously the content according to the user's interaction. For example, a concept is not shown unless another preliminary concept has already been accessed. Our approach presents both personalization mechanisms. As to the adaptable one, a distinguishable aspect w.r.t. other adaptable learning systems, is that in our case some choices are, generally, made by the student himself: reading style, layout properties and language; while still leaving the teacher free to take decisions regarding which kind of delivery criterion strategy has to be applied to that particular instance of the educational material (depending, as explained above, on pedagogical consideration). This two-fold adaptable mechanism augments the re-usable power of the system along with its flexibility.

3.1. Implementation choices

The choice of using XML technology to realize the proposed modelling approach, derives from four main considerations. Firstly, the intrinsic XML capability of defining a language via a Data Type Definition directly solves the problem of specifying the structure of the content of educational objects (e.g. in an analysis textbook the claim of a theorem must come before its proof) and of the overall textbook organization (chapters, paragraphs, subparagraphs, or folders, subfolders, files). Moreover the presence of a DTD along with the use of validating XML parsers helps the author to preserve the consistency of the content, at least, at structural level [18]. Secondly the capability of XSLT [19], to extract and re-elaborate any subparts from a source XML document to create a newly one with all the application and output-media dependent presentation features, being automatically generated, allows the system to retrieve and to reveal the diverse knowledge contribution brought by different weighted objects. Thirdly XLink [20] supplies standard navigational built-in-facilities, which allow us to directly represent the hierarchic structure of objects and forests by documents incorporating XLinked elements. The relevant characteristic of this mapping is that is-part relationships and semantic associations between resources, are dealt with by the same mechanism. However, due to the actual lack of any consolidated and widely available XLink processor (both at the client side and the server side) the management of XLink elements is done, in our system, by XSLT stylesheets which appropriately deal with the different kinds of links, traversal rules, roles, …. Notwithstanding this fact, we believe the proper use of XLink features to model the textbook structure (particularly the is-part relation), will be of benefit when standard compliant tools, such as browsers, will start to become more widespread. Finally the capability of XPointer [21] to address the internal structures of XML documents without having to pre-tag them, supplies a powerful way to accurately include external third-party (sub)resources thus augmenting the flexibility and encouraging the openness of the resulting hypermedia.

3.2. System architecture

The system delivers the course material as html pages accessed through an expandable index or by sequential navigation buttons. When the student asks for a page its content is dynamically created, at server side, starting from the objects pertaining to the course, and from the user data maintained in a relational database (specifically the open source MySQL). This process is supervised by the adaptation engine which coordinates the activity of several Java servlets whose main task is to process the diverse XSLT stylesheets against the proper XML documents in order to produce the personalized Web pages. The three main elaborations regard the creation of the three-framed course's interface, presenting the content, the course index and the navigation buttons.

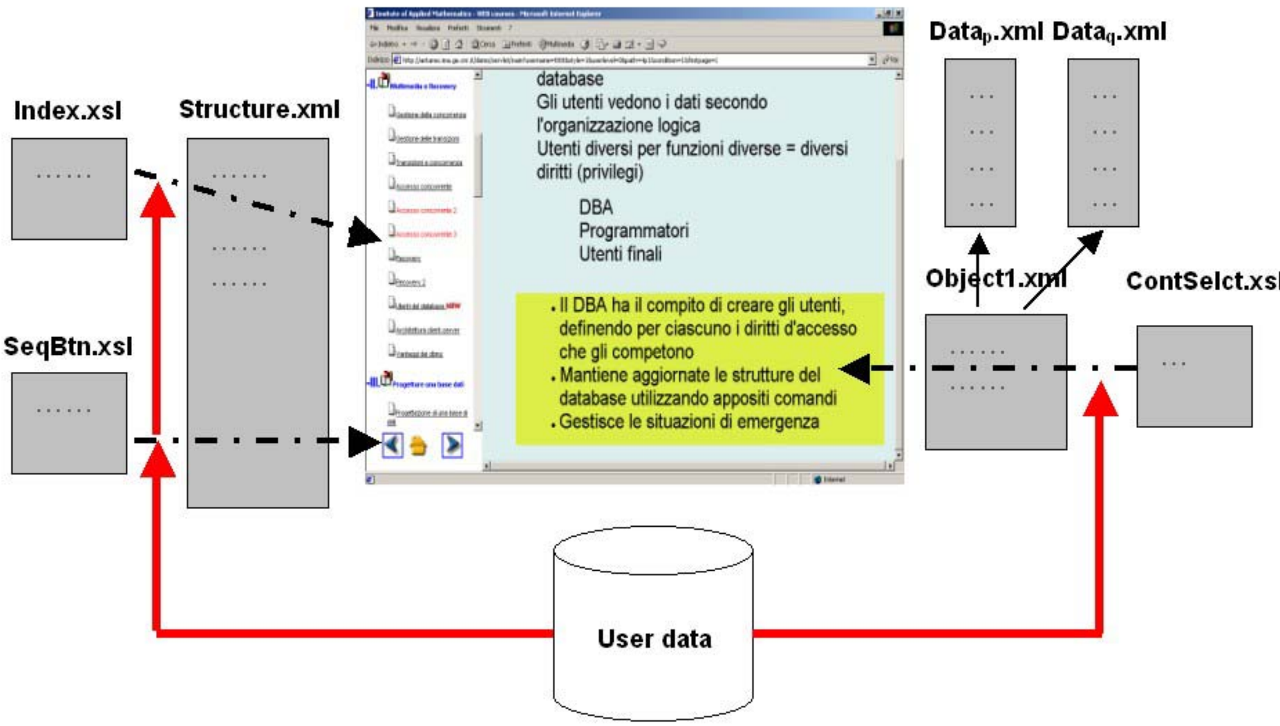


Fig. 3. Exemplification of the textbook generation process.

Following the schema in Fig. 3, the hierarchic index and the sequential buttons are obtained by applying two different XSLT stylesheets (respectively Index.xsl and SeqBtn.xsl) to the XML file containing the XLinked structure of the course (Structure.xml). The content page is similarly obtained by applying the ContSelct stylesheet to the XML file associated to the requested object (Object1.xml in this example). The main tasks of that stylesheet are the selection of the URI of the correct subcomponents, the retrieval of the content of these subcomponents from other documents (e.g. Data$_p$.xml, …, Data$_q$.xml) and the formatting of the fetched content according to the presentation properties (e.g. page layout, student preferred reading style,…). In all cases the servlets pass to the XSL processor (Xalan API) a built-on-the-fly XML document merging the XML static files along with the user collected information.

Other than to supervise the previous processes the adaptation engine constantly checks if the user has satisfied the current delivery condition, in order to update his knowledge state. To accomplish this task the engine compares the user activity, recorded in relational tables, with the delivery criterion contained in XML documents. As shown in (1) for each criterion the delivery conditions are listed. Each of them specifies the name of the delivery function along with the objects it applies on.

```
<criterion crtid=id crtdsc=description>
  <condition level="0">
   <function id="access" par="none"/>
   <cobject xlink:href="../introduzione.xml"/>
   <cobject xlink:href="../database.xml"/>
   <cobject xlink:href="../queries.xml"/>
  </condition>
  <condition level="1">
    <function id="access" par="none">
    <cobject xlink:href="../queries.xml"/>
    <cobject xlink:href="../security.xml"/>
  </condition>
</criterion>
```
(1)

A sketch of the monitoring and updating process undertaken by the system is reported in Fig. 4. As can be noted, to optimise system performance, only the activities related to the delivery model and specifically the conditions of level *l* are collected for a user with actual knowledge *l*, while all of his actions are checked against this set and removed from it if satisfying the underlying condition.

activity_records$_{(user,l)}$ ← {condition_of_level(*l*)}

While (activity_records$_{(user,l)}$ ≠ ∅)

If (action$_{(user)}$ ∈ activity records$_{(user,l)}$ and satisfies(action$_{(user)}$))

activity records$_{(user,l)}$ = activity records$_{(user,l)}$ \ action$_{(user)}$

level$_{(user)}$ = *l*+1 /* update user actual knowledge value */

Fig. 4. Monitoring and updating user knowledge state

[4] PERSONALIZATION STRATEGIES

It is commonly understood that the personalization effects in hypermedia systems, rely on three major personalization strategies: adapting content selection, adapting navigation support, and adapting presentation. In the following sections we analyse the behaviour of the adaptation engine according to these three personalization strategies, discussing implementation choices based on the use of XML technology.

4.1. Content selection

Two kinds of content extraction are operated by the adaptation engine. The first concerns the language of the student and intervenes at the adaptable level. The mechanism of language filtering is extremely straightforward, being realized through the use of the (standard) xml:lang attribute which is associated to each 'local' version of every subcomponent built by the authors, as shown in (2).

```
<bc name=bcname author=bcauthor>
    <bcl xml:lang="it">
        italian  local component content goes here
    </bcl>
    <bcl xml:lang="en">                                  (2)
        english local component content goes here
    </bcl>
    ………….
</bc>
```

The second and, from a pedagogical point of view, more relevant selection of content is based on the user actual knowledge. The dynamic page-generation mechanism, regarding an object O, is affected by applying the ContSelct stylesheet to the XML document associated to O. All subcomponents of O, through their corresponding XLink elements, are filtered according to the following XSLT template clause:

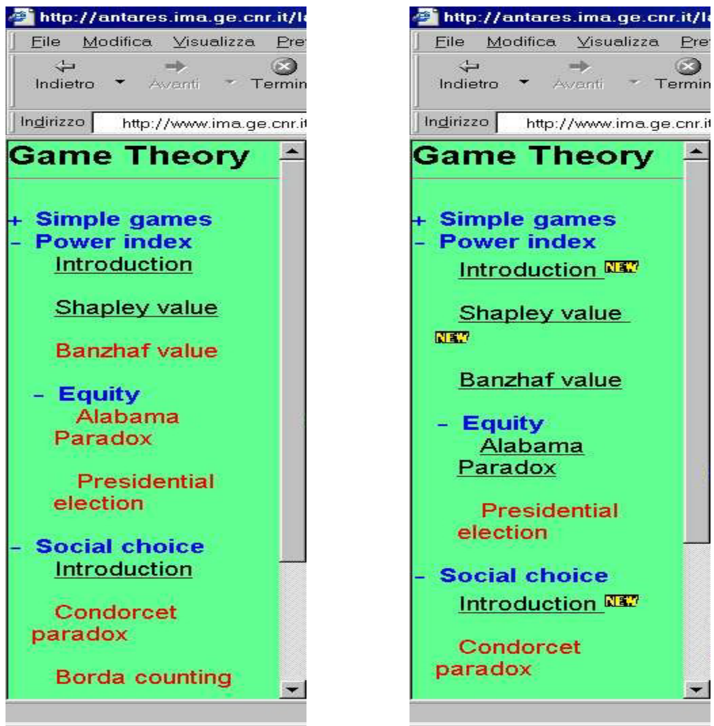
```
<xsl:template match="*[@xlink:href and @weight
      <= (//user/@state)]">                     (3)
```

The matching rule in (3) states that only the subcomponents with weight less than or equal to the actual state of the user, are to be selected, via the URI-value of the xlink:href attribute, and shown to the user. It is worth noting that the choice of which content shown to the user, i.e. the already visited along with the new one as in this case or only the new one (using respectively the less-than-equal relation or the equality relation), may be determined case by case by the author, so supplying another way to adapt content fruition.

4.2. Adapting navigation

Web surfers are general acquainted with three main navigation techniques: maps or indices, sequential navigation buttons and text-embedded hyperlinks. In a adaptive hypermedia the user should explore only her personalized portion of the navigation space. To this end the above techniques have to be reviewed.

In our system the main entrance to course material, is supplied by the personalized index which reflects the user tailored hierarchical structure of the course. The index restricts the navigation space to the accessible pages, displaying as normal text (that is, disabling) links to the pages which are not accessible and colouring them differently (red in Fig. 5a). When the state of knowledge of a student progresses, due to his success in the satisfaction of some criterion, the index is update by the adaptation engine reflecting two main kind of changes in the explorable content. The first one concerns the new objects that become visitable, i.e. which weight equalised the new knowledge state of the user. For these objects the relating links in the index are ‘switch on’ (underlined black in Fig. 5b). The second change is related to previous hidden components of already accessed object, for example the subcomponent bc8 of object O3 of course A in Fig. 2 is not shown to a student with actual state of 0. In this case to highlight that something new has became visible for an already visited object, the icon “New” is set near the link in the index (i.e. Fig.5b).



a) **b)**

Fig. 5. Index updates reflecting user knowledge deepening

The second way to navigate the content domain is through the forward and backward buttons, which are generated on the basis of the textbook structure by applying to it the SeqBtn stylesheet. To determine the two objects visitable by the student, given the current visualized object, which precede and follow it, condition (4) is applied by the system:

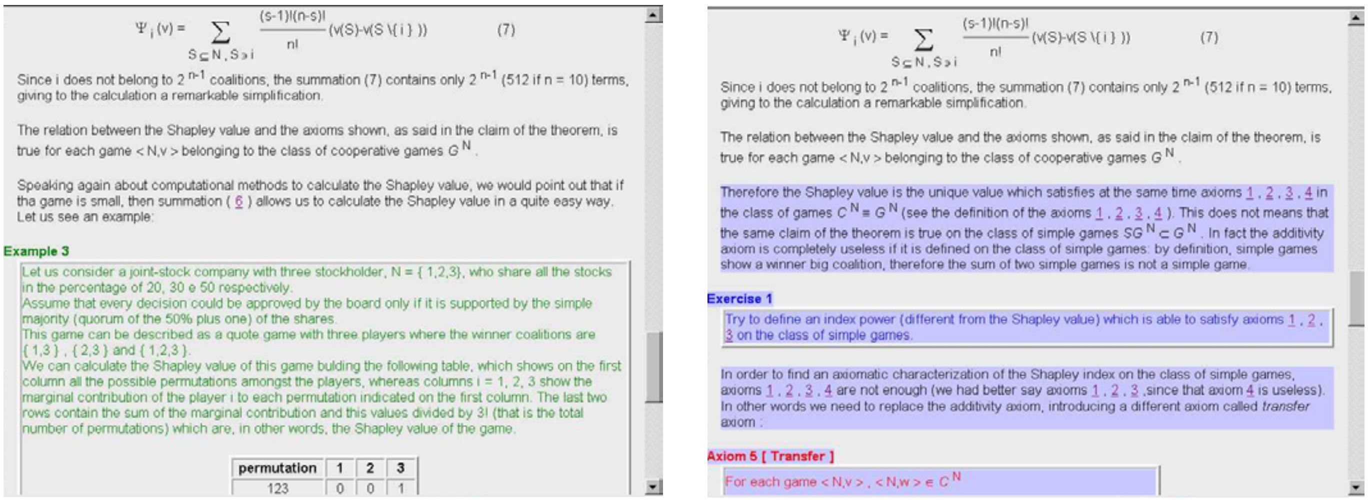
```
<xsl:if test =
    "boolean(./preceding::cobject[@weight<=
    //user/@state"])">
```
(4)

Using the XPath `preceding` axis [22], the system easily backwards navigates the tree structure of the XML document, searching for the nearest preceding object with a weight less than or equal to the user state. If such an object exists its URI is associated to the backward arrow. In a symmetric way the system implements the forward button by using the `following` axis.

As to the traditional text-embedded hyperlinks, they are treated similarly to the previous cases. In fact, as said in section II.A, a hyperlink, possibly to some external resource, is anyway a subcomponent and has a weight. According to that weight the link is shown or disabled.

4.3. Content presentation

Other than the previous techniques reflecting the changes occurring in the navigational space, content presentation accounts for informing the student about the semantic differences resulting at content level. Our system envisages three principal presentation mechanisms. Firstly, to make the student aware of his progress from a knowledge state to the successive state, and hence to best reflect the diverse cognitive contributions of different weighted subcomponents of an object, the system highlights them with different background colours (i.e. the cyan background text in Fig. 6b).



a) b)

Fig. 6. Diferent views of the same page for students with different level of knowledge

A second kind of content presentation technique is used to capture the different types of subcomponents belonging to an object. For example a page of mathematics usually contains general explanations, definitions, some theorem or lemma, examples, exercises, and so forth; while a page devoted to literature usually contains some biographic notes about a writer, a list of publications, the influence of his work, some criticism, bibliography, related authors,… To emphasize this diverse semantic contribution, we think it valuable to give the author the choice to differently colour each type of content (for example "Exercise" and "Axiom" are blue and red in Fig 6b while "Example" is in green in 6a). Thirdly, the system supplies the reader with the feature of stretching-shrinking the text, by clicking on the name of the subcomponent used as a placeholder, that as noted by [23] allowing the student to alternatively switch on-off the part of the content that at any moment he considers more or less relevant.

All the previous presentation features are realized by the ContSelct stylesheet that for each retrieved subcomponent constructs, through XSL template rules, all the required html tagging and associates the most appropriate scripting code (e.g. to realize stretching-text functionalities), based on the teacher decisions and student preferences. Again, it is easy to see how these presentation techniques supply a further kind of re-use mechanism to hypermedia authors.

[5] Conclusion

Clearly the process of building a personalizable system is normally more costly than that of a traditional one. As shown in this paper, two main factors impact on the production of tailorable hypermedia. The amount of personalized material to be realized is normally greater than one of a general-purpose one. Moreover, the user characteristics have to be modelled to allow the tailoring mechanism to properly function. However, our initial experiments along with literature results [24], [25], seem to show that carefully designed personalizable educational material may improve learning activities while allowing the exchange of content between educational operators and institutions.

REFERENCES

[1] Keller D. and W. Schaufelberger; *Use of the WWW in Control Engineering Education; Engineering Education: Redescovering the Centre*. Proceedings of SEFI Annual Conference '99, Winterthur and Zurich, Switzerland, 1999

[2] *Instructional uses and effects of World Wide Web course pages: A Review of instructor experiences*. The American Communication Journal, ISSN: 1532-5865, Vol. 1, Issue Three (1.3), 1998

[3] Martin, J., *The role of consortia in educational broadcasting*. Australian Journal of Educational Technology, 2(2), pp. 94-97, 1986.

[4] http://ltsc.ieee.org.

[5] http://www.imsproject.org.

[6] http://www.adlnet.org.

[7] de Bra P., Brusilovsky P.,Conejo R. (Eds.). Proceedings of Adaptive Hypermedia and Adaptive Web-Based Systems Second International Conference, AH 2002 Malaga, Spain, 2002.

[8] Chittaro, L. and Ranon, R., *Dynamic Generation of Personalized VRML Content: a General Approach and its Application to 3D E-Commerce*. Proceedings of Web3D 2002: 7th International Conference on 3D Web Technology, ACM Press, New York, February 2002, pp. 145-154

[9] Steiner, K.E., & Tomkins, J., *Narrative event adaptation in virtual environments*, Proceedings of the 2004 International Conference on Intelligent User Interfaces, Madeira, Portugal, 2004.

[10] O'Grady, M.J. & O'Hare, G.M.P., *Enabling Customized & Personalized Interfaces in Mobile Computing*, Proceedings of Intelligent User Interfaces, , January, Island of Madeira, Portugal, ACM Press, pp. 292-294, 2004.

[11] Gates, K. F., Lawhead, P. B., Wilkins, D. E., *Toward an adaptive WWW: a case study in customized hypermedia*, New Review of Multimedia and Hypermedia 4, pp. 89-113, 1998.

[12] Park, O., & Lee, J., *Adaptive instructional systems*. In D.H. Jonassen (Ed.) Handbook of Research for Educational Communications and Technology (pp. 651-685). Mahwah, NJ: Erlbaum., 2003.

[13] Wu, H., De Kort, E., De Bra, P., *Design Issues for General-Purpose Adaptive Hypermedia Systems*, Proceedings of the ACM Conference on Hypertext and Hypermedia, pp. 141-150, Aarhus, Denmark, 2001.

[14] Connolly D., *Naming and Addressing: URIs, URLs,...*, http://www.w3.org/Addressing.

[15] Henze H. and Nejdl W., *Extendible Adaptive Hypermedia Courseware: Integrating Different Courses and Web Material,* International Conference on Adaptive Hypermedia and Adaptive Web-based System, LNCS 1892, Springer-Verlag, Berlin, pp. 109-120, 2000.

[16] Brusilovsky, P. (1996). *Methods and Techniques of Adaptive Hypermedia*. In: User Modeling and User-Adapted Interaction 6: 87-129, Kluwer academic publishers, 1996.

[17] Kobsa, A., Koenemann, J., Pohl, W., *Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships*. The Knowledge Engineering Review 16(2), pp. 111-155, 2001.

[18] http://www.w3.org/TR/2000/REC-xml-20001006.

[19] http://www.w3.org/TR/xslt.

[20] http://www.w3.org/TR/xlink.

[21] http://www.w3.org/TR/xptr.

[22] http://www.w3.org/TR/xpath.

[23] Boyle C., Encarnacion A.O., *Metadoc: An Adaptive Hypertext Reading System*, in User Modeling and User-Adapted Interaction 4, 1-19, 1994.

[24] Hock, K. (1997) Evaluating the Utility and Usability of an Adaptive Hypermedia System, Proceedings of 1997 International Conference on lntelligent User Interfaces, pp. 179-186. ACM, Orlando, Florida, 1997.

[25] Hothi, J. & Hall, W. (1998). "An Evaluation of Adapted Hypermedia Techniques using Static User Modelling". In Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, Pittsburgh, USA, pp 45-50, 1998.