Personalization of User Interfaces for Browsing XML Content Using Transformations Built on End-User Requirements

Benoît Encelle
Generalized Information Systems
IRIT – University Paul Sabatier
Toulouse, France
encelle@irit.fr

Nadine Baptiste-Jessel
Generalized Information Systems
IRIT – University Paul Sabatier
Toulouse, France
baptiste@irit.fr

ABSTRACT
Personalization of user interfaces for browsing content is a key concept to ensure content accessibility. In this direction, we introduce concepts that result in the generation of personalized multimodal user interfaces for browsing XML content. Users requirements concerning the browsing of a specific content type can be specified using user-friendly description languages. According to these specifications, transformation rules are generated in order to produce personalized user interfaces for browsing specific content types. With the emergence of the semantic Web and connected XML applications, such customized multimodal user interfaces can be useful for many kinds of users, especially individuals with various type of impairment.

Categories and Subject Descriptors
H.5.2 **[Information Interfaces and Presentation (e.g., HCI)]**: User Interfaces – *User-centered design.*

General Terms
Design, Human Factors, Theory.

Keywords
Adaptable User Interfaces, User-interfaces for the elderly or disabled, Model-based user interfaces, Transformational approaches.

1. INTRODUCTION
Currently, content providers mainly drive the presentation of content and the end-user navigation (i.e. scanning) possibilities inside this content. As a result, user interfaces for browsing this content - with common web browsers – remain almost identical, using the same modes of interaction whatever the user.

However, people with special needs (e.g. visually impaired users or highly mobile individuals) that cannot access Web content using traditionally used modes of interaction (e.g. visual mode) require customized presentation and scanning possibilities with this content.

For instance, in order to easily access available information, blind users need a fine-tailored multimodal browsing system that takes advantage of using several modalities/media (e.g. a text-to-speech synthesizer and a Braille display) for communicating the content.

Moreover, users may have, according to the content type, very different expectations in terms of, for instance, how information has to be presented [3].

As a result, we develop a model for specifying end-user requirements concerning the presentation and scanning possibilities for a content type. This specification of the end-user "browsing" requirements for a particular content type results in the generation of a set of corresponding transformation rules. The application of these transformation rules to an XML-conformant content leads to the generation of an adapted end-user interface for browsing this kind of content.

In this paper, we firstly discuss related work (section 2). In section 3, we introduce the model and concepts we developed for specifying end-user requirements concerning the browsing of specific content types. We develop in section 4 the model for applying it on XML content and describe two representations of the end-user browsing requirements: a user-friendly one and a system-friendly one. In section 5, the transformation process, from the end-user browsing requirements to the generation of personalized user interface is explained. In section 6 we present the results of utility and usability evaluations of the proposed concepts. Implementation highlights of the system we developed for generating adaptable user interfaces are provided in section 7. Finally, we conclude in section 8.

2. RELATED WORK
2.1 Web Content Accessibility
In order to make Web content accessible, transformations (i.e. adaptations) of content must be performed. We distinguish two kinds of adaptations: adaptation performed at authoring-time (i.e. during the writing of documents) and adaptation performed at run-time (i.e. during content consultation).

2.1.1 Authoring-time Content Adaptation
Authoring-time adaptation firstly results in a set of "accessibility" rules/guidelines that developers/authors of authoring tools, user agents/Web documents have to follow. In this way, the Web Accessibility Initiative (WAI) [22] from the W3C wrote a set of guidelines.

Moreover, for visually impaired users or highly mobile individuals, audio browsable Web content have been developed. For this purpose, several markup languages such as VoiceXML, SALT, XHTML+Voice and voice browser systems have emerged.

Although WAI and development of specialized languages are very promising and needed, adapting content using guidelines or adapting content to voice browser technology still remains a significant burden for many content providers and has a significant cost.

W4A2007 - Technical Paper, May 07–08, 2007, Banff, Canada. Co-Located with the 16th International World Wide Web Conference.

2.1.2 Run-Time Adaptation

The second kind of adaptation (Run-Time Adaptation) consists in preserving Web content as it is and results in the development of assistive technologies (e.g. screen readers, JAWS) or Web browsers dedicated to people with special needs (e.g. HearSay [23], IBM HomePageReader). Using these software, the user browses content though multiple modalities. However, most assistive technologies and dedicated Web browsers do not provide to the end-user sufficient filtering options for eliminating unwanted information [23][1]. Generally speaking, due to the fact that this kind of software is already "dedicated" to a well specialized group of users, the personalization or customization aspect (i.e. the adaptation to each user's needs) is often a bit limited. An interesting research area focused on the personalization of content is Adaptive Hypermedia.

2.2 Content Personalization

According to Brusilovsky [24], Adaptive Hypermedia (AH) is one of the most promising application areas for user modeling and user-adapted interaction techniques. AH systems (AHS) build a model of the context in which they are used. For instance, information concerning the user (goals, knowledge, preferences), the environment (available devices), is included in this context model. Adaptive means that the system permanently collates information about the user into the model and automatically applies it for customizing content and navigation support.

In our point of view, the adaptation performed by adaptive hypermedia is more an adaptation at content level rather than an adaptation at the level of presentation and interaction possibilities with this content. That's why Adaptive Hypermedia Systems often are on-line educational systems, personalizing content according to user skills and user knowledge about the information topic.

3. THE MODEL OF POLICIES/PROFILES

3.1 Identification of Requirements

For determining personalization requirements for content browsing we used [1] as a reference study. Only requirements for content reading and scanning possibilities (i.e. navigation possibilities inside content) are studied.

3.1.1 End-user Personalization Requirements

According to [1], the following communication problems between the system and the end-users and deduced personalization requirements have been identified:

3.1.1.1 From the system to the end-users (presentation of content)

- Information overload

Requirement 1 (**R1**): For each kind of content, user must have the ability of choosing sub-information that have to be really presented by the system.

- Inadequate output modalities

Requirement 2 (**R2**): For each kind of content that has to be presented, user must have the ability of choosing the right output modalities.

3.1.1.2 From the end-users to the system (navigation/ scanning possibilities inside content)

- Inadequate navigation possibilities

Requirement 3 (**R3**): For each kind of content, user must have the ability of specifying inside-content navigation / scanning possibilities.

- Inadequate input modalities

Requirement 4 (**R4**): For each navigation (i.e. scanning) possibility, user must have the ability of choosing the right input modalities.

3.1.2 Deduced Requirement

According to R2 and R4, an adaptation of communication modalities/media between the end-user and the system is required. As a result, the content has to be adapted to a media-readable format in order to be transmitted. (**R5**)

3.2 Policies/Profiles Concepts

In order to represent preceding identified requirements, we introduce the concept of profiles of policies.

3.2.1 Profiles

In order to personalize system reactions to each user, individual needs or preferences are gathered into a *profile* [4]. The usage of profiles reduces the 'one-size-fits-all' approach to a 'one-to-one' approach [5]. In order to separate the profile components that can be described by the end-user him-self and those that have to be specified by an expert, we suggest the following architecture (c.f. figure 1).

According to this architecture, an entity called User Global Profile is made up of the specifications of all personalization requirements. This architecture puts into light several classifications of profile components.

3.2.1.1 Stereotype-based and Personalized Profiles

User adaptable or adaptive systems like Adaptive Hypermedia Systems (A.H.S)[6] often initialize a user profile using a stereotype-based profile [7]. A stereotype-based profile corresponds to average preferences of a set of users that share common characteristics [8]. For instance, in the AVANTI Project, there are several stereotyped profiles that represent the interests of dystrophic users, wheelchair users and blind users [9]. The main objective of the AVANTI project was to demonstrate that it is possible to develop generic multimedia telecommunications applications, which are adaptable and adaptive to the requirements of most potential users including disabled people, elderly people, occasional users and other users whom we like to term 'professionals'.

Using stereotype-based profiles, the user takes directly advantage of the system without having to set up all his preferences. Stereotype-based profiles have to be set up by experts that have a good knowledge of a specific disability (e.g. blind institutions members). A stereotype profile only represents preferences tendency: because individual needs and requirements are diverse and can be difficult to anticipate [10], a personalized profile, gathering user own preferences is needed too.

For a given kind of content, preferences described into a personalized profile override the corresponding preferences of a stereotype-based profile.

3.2.1.2 Content and Media Profiles

All identified user personalization requirements (c.f. section 3.1.1) can be specified by the end-user himself. However, the inferred requirement R5, namely "*the adaptation of the content to a media-readable format*" rather requires the technical skills of an expert.

As a result, requirements specifications are split up into two distinct profiles: a content and a media one.

3.2.2 Applications Policies

3.2.2.1 Definition

In previous works [11][12][13], we introduce the concept of application policies. This concept was developed in order to fulfill previous enunciated requirements.
A short definition of this concept could be the specification of the system's reaction according to a given user event/action and a given kind of content. According to this definition, an *Application Policy* is connected to a specified kind of content (e.g. XML applications: XHTML, MathML, RSS, etc.).

3.2.2.2 Classifications of applications policies

Profiles/policies architecture is presented in figure 1.

Task-oriented specialization of applications policies

Applications policies have to result in the end-user customization of the following *"browsing tasks"*:

- presentation of content
- navigation possibilities inside content (scanning possibilities)

User preferences in terms of presentation and navigation possibilities are then separated into two kinds of application policy: presentation policies and navigation policies.
As a result, application presentation and navigation policies extend the concept of *Application Policy*:

1. Application Presentation Policies

   An application presentation policy (e.g. RSS presentation policy) specifies how the system has to present a particular kind of content (e.g. RSS content).

   An application presentation policy thus describes two components for each type of content element (or set of elements) that should have particular presentation intents:

   ➢ Firstly, which content parts (sub-elements) must be presented to the user: this is the idea of filtering the content. (R1)

   ➢ Secondly, which modality/media present these content parts (link between the part and the modality/media) and description of the presentation order in time of these content parts (i.e. multimedia/multimodal choreography description). (R2)

1. Application Navigation Policies

   A navigation policy describes two components for each type of content element (or set of elements) that should have particular scanning/navigation intents (R3-R4):

   ➢ Firstly, the specification of the user actions that produce navigation actions. A user action is a condition based on user events (i.e. a key pressed event, mouse click, a sentence pronounced by the user and so on). (R4)

   ➢ Secondly, for each user actions, the targeted content element that has to be presented. (R3)

Media-specific applications presentation policies

Media-specific applications policies are used to indicate to a particular media how a particular content has to be processed (R5). A media-specific application presentation policy describes the logic for adapting content encoded with this application to the media (i.e. formatting rules). For instance, a Text-To-Speech synthesizer presentation policy for XHTML content will specify the prosody information that have to be applied during the vocal presentation of XHTML elements Title, H1 and so on.

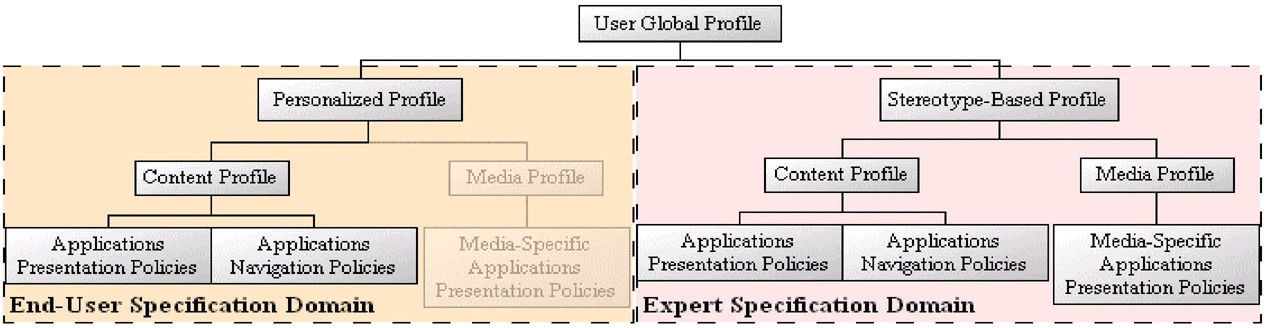

Figure 1. Architecture of profiles/application policies.

3.3 Algorithm for Selecting the most Adapted Preferences

For a particular usage context *C* (a given content *E* and a given task (presentation/navigation)) (e.g. presentation of the XHTML element Title), the user preferences that have to be applied to C are selected using the following algorithm:

1 Does a dedicated policy in the personalized profile match C? → Apply it

2 (else) Does a default policy (i.e. for any kind of XML content) in the personalized profile exists ? → Apply it.

3 (else) Does a dedicated policy in the stereotype-based profile match C ? → Apply it.

4 Apply the default policy of the stereotype-based profile

4. POLICIES REPRESENTATION FOR XML CONTENT

4.1 Relations between policies components and XML technologies

XPath [14] is used to target XML element (or set of elements). As a result, XPath expressions are used for specifying components R1 and R3.

For choreographing content presentation (R2), specific features of SMIL [15] taken from the "SMIL Timing and Synchronization Modules" are used.

4.2 End-User Representation

In order to give to the user the possibility of writing their own applications policies, two description languages - one for each kind of application policy – have been developed using LL(1) grammars [16] written in JavaCC [17].
Using these description languages, the user could describe policies components that integrate his profile.
For further details concerning these description languages, the reader can refer to [13] [18].

Example of user preferences specification for RSS Content

- RSS content simplified schema

An *rss* element contains one or more *item* element. Each *item* element corresponds to particular "story", described using its title (*title* element), its synopsis (*description* element) and its publication date (*pubDate* element).

- Specification of RSS examples policies using description languages

For the following examples, we assume that a Text-To-Speech synthesizer (TTS), a Braille display device and an Automatic Speech Recognizer (ASR) are available on the XML browsing system of the end-user.

➢ Presentation policy for the *rss* element type

rss **:** *item* **[(** *title***:TTS ,** *description***:BrailleDisplay)];**

Legend:

italic: XPath expressions,

bold: choreography information:
- X**:**Y**;** : means that the content part X has to be presented using the presentation intents Y. (**c1**)
- X**[**Y**]** : means that each X elements will be presented in sequence with the presentation intents specified in Y. (**c2**)
- **(**X_1 **,** …**,** X_n**)** means that elements X_1 until X_n will be presented in sequence. (**c3**)
- **(**X_1 **|** **…** **|** X_n**)** means that the elements X_1 until X_n will be simultaneously presented (not used in the example). (**c4**).
- X**:**Y means that the content part X has to be presented using the media/modality Y. (**c5**)

Explanation: presentation policy for an "*rss*" element: for each child "item" element, its "title" element is presented using the TTS, following with the displaying on the Braille line of its "description" element. pubDate element is not presented (i.e. filtered, c.f. R1).

➢ Navigation policy for the *rss* element type
rss:**ASR.Rule=Number**:*item[ASR.BestWord]*;

Legend: *italic*: XPath expressions, **bold**: user action specification

Explanation: when the system presents an "rss" element, if the user pronounces a number (i.e. N), the system navigates to the "item" element number N (if exists) and presents it.

4.3 System Representation

In a system point of view, presentation/navigation applications policies can be represented using transformation rules (XSLT [19] templates).
Conversion modules have been developed to switch between these two representations of policies.

5. FROM THE END-USER REQUIREMENTS TO THE GENERATION OF ADAPTED USER INTERFACES

5.1 From the End-User Representation to the System Representation of Policies

The parsing of the end-user representation of an application policy generates its corresponding system representation (i.e. transformation rules).
LL parsers were developed to handle end-user representation of policies. The parsing of the end-user representation of a policy firstly generates a syntax tree. Once this tree of objects is built, the generation of the corresponding system representation (i.e. transformation rules: XSLT templates) is carried out with the help of the visitor design pattern [25]. The visitor pattern enables the definition of a new operation on an object structure without changing the classes of the objects. As a result, syntax tree operations (i.e. XSLT code generation) can be written as so-called visitors without changing and recompiling the syntax tree node classes.
For instance, a simplified version of the associations between the nonterminals of the presentation policies grammar (i.e. the classes of nodes of the syntax tree) and the corresponding generated XML elements (XSLT elements/SMIL-like elements) are shown in the table 1.

Table 1. Associations between the nonterminals of the Presentation Policies Grammar and corresponding XML elements.

Nonterminals	Associated XML element
Element type preferences - X**:** …**;** (c1)	xslt:template match="X"
Loop - X[…] (c2)	xsl:for-each select="X"
Sequence - (X1**,** … **,** Xn) (c3)	SEQ
Parallel - (X1 **|** … **|** Xn) (c4)	PAR
Media – X**:**Y (c5)	MEDIA name="Y" xslt;copy-of select="X"

Example

In order to illustrate the conversion process from an end-user representation of policy to its corresponding system representation, we consider an example of a presentation policy for the *rss* element type.

- *End-User Specification of the Policy (c.f. section 4.2 for details)*
  rss : item [(title:TTS , description:BrailleDisplay)];

- *Policy Syntax Tree*
  The syntax tree corresponding to this policy is illustrated in figure 2. Tree nodes correspond to grammar nonterminals and tree leaves are grammar terminals.

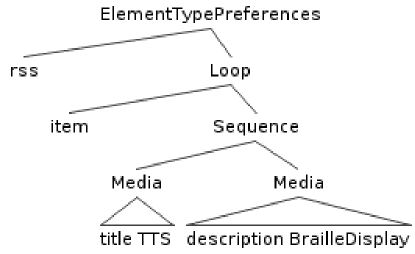

Figure 2. Syntax Tree of the Example of Presentation Policy

- *System representation of the policy (XSLT Template)*
  The visit of the syntax tree using suitable associations (c.f. table 1) produces the following XSLT Template:

```
<xsl:template match="rss">
  <SEQ>
  <xsl:for-each select="item">
   <MEDIA name="TTS">
    <xsl:copy-of select="title"/>
   </MEDIA>
   <MEDIA name="BrailleDisplay">
    <xsl:copy-of select="description"/>
   </MEDIA>
  </xsl:for-each>
 </SEQ>
</xsl:template>
```

5.2 From the System Representation of a Policy to the User Interface Generation

In order to illustrate this process, we consider the following situation: the end-user wants to browse an RSS document.
As the root element of all RSS documents is an *rss* element type, the XML browsing system has to "present" this content type to the end-user and then retrieves the system representation of the most adapted application presentation policy (**pp**) from the User Global Profile that can handle this *rss* element, according to the algorithm described in section 3.3.

5.2.1 Presentation of the Content to the End-User

In order to retrieve the end-user presentation preferences concerning the *rss* element that has to be shown, an XSLT transformation on the *rss* element is performed using transformation rule (i.e. XSLT Template) of *pp*.
The resulting document corresponds to the description of the presentation intents for the *rss* element. The structure of this document is the following:

```
<SEQ>
 <MEDIA name="TTS">item1_title</MEDIA>
 <MEDIA name="BrailleDisplay">item1_description</MEDIA>
 . . .
 <MEDIA name="TTS"> itemn _title </MEDIA>
 <MEDIA name="BrailleDisplay">itemn_description</MEDIA>
</SEQ>
```

The interpretation of this document (using a SMIL-like player) results in the adapted presentation of the *rss* element to the end-user.

5.2.2 Handling End-User Navigation/Scanning Actions

During the presentation of the *rss* element, if the user performs actions (user events on input devices), these events are represented as a list of XML attributes and are added to the *rss* element.

As an example, if the user pronounces the number "2" during the presentation of the *rss* element, the following attributes are added to the element:

```
<rss ASR.rule= 'Number'ASR.BestWord='2'>
 ...
</rss>
```

The XML browsing system will then retrieve the system representation of the most adapted application navigation policy (**np**) that can handle this *rss* element, according to the algorithm described in section 3.3.
The corresponding XSLT template is the following:

```
<xsl:template match="rss[@ASR:Rule='Number']">
 <xmb:target>item[<xsl:value-of select="@ASR:BestWord"/>]
</xmb:target>
</xsl:template>
```

Next to that, an XSLT transformation using the transformation rule of *np* is applied on the *rss* element.
The resulting document is the following:

```
<xmb:target>item[2]</xmb:target>
```

Interpretation: the xmb:target element contains the XPath expression that targets the next content that has to be presented to the end-user (R3).
This XPath expression is then evaluated according to the *rss* element (contextual node). As a result, in regards to the given example, the XPath expression (i.e. item[2]) targets the second *item* element of the *rss* element. The XML browsing system now has to present this *item* element. The most adapted presentation policy is retrieved and so on.

6. EVALUATION

6.1 Discussion

6.1.1 Evaluation Context

Generally speaking, evaluation of an interactive system in the HCI field consists in measuring its utility and its usability. To summary, the utility of a system generally refers to the fact that it allows the user to complete relevant tasks. The usability of a system refers to the ease and efficiency within its user interface can be operated.
As the proposed concepts tend to generate, according to user preferences (i.e. profile of policies), a personalized user interface for browsing XML content, measuring the utility and usability of a specific generated user interface does not seem to be relevant.
As a consequence, the evaluation of our proposition consists in measuring the utility and usability of the mechanisms for describing user interfaces (i.e. mechanisms for describing policies).

6.1.2 Evaluation Dimensions

6.1.2.1 Utility

Firstly, the expressivity power of policies mechanisms for describing user interfaces has been estimated. The goal is to know the kinds of user interfaces that can be generated using these mechanisms.

6.1.2.2 Usability

Secondly, because the end-user must have the possibility to create or change easily and quickly its preferences (its user interfaces), usability of policies description languages/mechanisms has been measured.

6.2 Principles and Results

6.2.1 Utility evaluation

6.2.1.1 Principle

Generally speaking, the goal of these generated user interfaces is to improve content accessibility in using multimodal communication possibilities.

As a result, utility evaluation consists in qualifying the kind of multimodal interactions that can be described into these generated user interfaces. To qualify the kind of cooperation between modalities that such generated user interfaces can support, the CARE Framework and properties [20] are used. The CARE properties define a framework for characterizing different forms of multimodal usages of an interactive application.

6.2.1.2 Results

Results interpretation leads to the conclusion that all CARE properties, except one, can be instantiated using the policies description languages.

For having details concerning the utility evaluation, the reader can refer to [13].

6.2.2 Usability evaluation

6.2.2.1 Principle

In order to evaluate the usability of policies description languages, usability of each language (presentation/navigation) has to be measured. Our goal is to know if any a priori novice end-user is able to use the description language.

Usability tests have been performed, asking the end-users to realize a set of representative tasks.

The ease of use of a description language has been quantified according to the following hypothesis: a language can be described as easily usable if two sets of users, one initially tagged as novice and the other initially tagged as expert, handle the language after X usages with roughly the same ease of use (X must be quite small).

Three exercises (i.e. task: the writing of an application policy) have been given to 20 subjects (10 experts/ 10 novices).

A short tutorial (15 min.) concerning the writing of application policies was presented to the subjects before the first exercise.

6.2.2.2 Parameters

Subjects

In order to maximize the probability of having expert and novice subjects, we made the following assumption: "someone who has programming skills has more chance to be an expert in the writing of application policies than someone who does not know anything about programming". As a result, subjects were students with different knowledge of programming. However, expert and novice groups of subjects have been built upon the results of the first exercise.

Tasks

Each exercise (i.e. task) consists, according to a given textual description of an end-user "browsing preferences" for a specific application (e.g. RSS), in the writing of the corresponding policy. The objective is to evaluate the subject understanding and its ease of use of the different concepts of a policy description language, through its writing of the different syntactic constructions of this language. Concerning the presentation policy description language for instance, the following concepts were identified (cf. section 3.1.1.1 for corresponding end-user requirements *R* and section 4.2 and table 1 for corresponding syntactic constructions *c*):

- Declaration of presentation intents for an element type (R1-R2, c1)
- Specification of content filters (R1 – c3, c4)
- Coordination of the presentations of parts of content over time (R1-R2 – c3, c4, c5)
- Declaration of looping constructs (part of R1, c2)

6.2.2.3 Results

Table 2. Usability Evaluation Results

Marks (/6) and %	**1st Exercise**	**2nd Exercise**	**3rd Exercise**
Expert: average	5.800	5.925	5.960
Expert: std. deviation	0.209	0.112	0.089
Novice: average	2.400	5.150	5.875
Novice: std. deviation	1.353	0.652	0.125
Averages difference	3.400	0.775	0.085
Ratio* (%)	**100**	**22.800**	**2.500**

According to data analysis, difference between the expert group average and novice group average tends to decline progressively.

For the final exercise, the ratio* (2.5%) seems to be trifling: distinction between a novice user and an expert one is increasingly difficult to carry out.

Moreover, standard deviation by groups decreases progressively: for each group, the level of users tends to become increasingly homogeneous.

The initial question was: "*Is any a priori novice end-user able to use a policy description language?*". According to assumption we made, the conclusion is that a novice subject is quickly able to employ a policy description language because it tends to become an expert only after three usages of the language.

7. IMPLEMENTATION

An Xml browsing system (c.f. figure 3) taking into account all the features of profiles of policies was developed using Java in order to be platform independent.

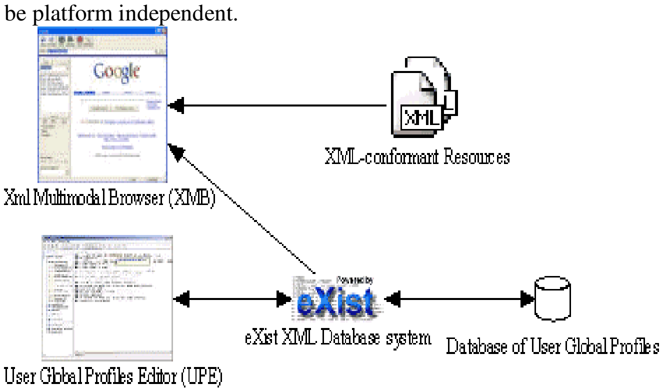

Figure 3. Components of the Xml content browsing system.

eXist [21] is the XML database system that stores the collection of User Global Profiles.

7.1 Xml Multimodal Browser (XMB)

The Xml Multimodal Browser (XMB) is the core component of the browsing system. After a user login step, XMB gets through the database the corresponding User Global Profile. When the user requests an XML-conformant resource (via its URL), XMB applies profile policies to personalize content presentation and navigation/scanning possibilities inside this content.

* Difference between expert and novice averages for the x^{nd} exercise divided by the same difference for the 1^{st} exercise.

7.2 User Global Profiles Editor (UPE)

The User Global Profiles Editor is used to create/modify user global profiles. A command-line version of the editor is also available for visually impaired users. User global profiles are created/updated into the database using eXist.

8. CONCLUSION

People with special needs (e.g. blind people) require fine-tailored multimodal browsing user-interfaces to easily access content. These user interfaces must be personalized according to the end-user preferences/requirements and according to the content type.

In this article, we firstly introduce briefly end-user requirements in terms of content "browsing" and suggest a model of concepts for representing them.

The concept of application policy we introduce results in the representation of end-user preferences in terms of presentation and navigation (scanning) possibilities concerning content encoded using a particular application (e.g. XHTML, MathML, RSS, SVG and so on).

The transformation process from a user-friendly representation of end-user browsing preferences to its corresponding system applicable representation is explained in details. Using this kind of representation, personalized user interfaces for browsing content can be generated on-the-fly.

Utility of application policies for generating adapted multimodal user interfaces has been evaluated. Moreover, the usability of the policies description mechanisms has been estimated too. On one hand, results of these evaluations show that these generated user interfaces can practically support every kinds of cooperation between modalities. On the other hand, policies description languages, according to usability tests we perform, are user-friendly (e.g. simples and quickly usable).

Personalizing user interfaces for browsing content using multimedia/multimodal possibilities is very interesting not only for disabled people but also for highly mobile individuals or older people. Moreover, due to the emergence of the Semantic Web[2], dedicated applications frequently appear (e.g. MathML, MusicXML, CML, RSS, XMLTV, etc.): the use of such policies results in the rapid development of browsing user interfaces for these applications content.

9. REFERENCES

[1] M. F. Theofanos and J. G. Redish, "Bridging the gap: between accessibility and usability", in interactions, vol. X6, ACM Press, 2003, pp. 36-51.

[2] W3C, Semantic Web. http://www.w3.org/2001/sw/

[3] Waltraud Schweikhardt, Cristian Bernareggi, Nadine Jessel, Benoît Encelle, Margarethe Gut, "LAMBDA: A European System to Access Mathematics with Braille and Audio Synthesis", Lecture Notes in Computer Science, Springer, vol. 4061, ICCHP'06: International Conference on Computers Helping People with Special Needs, Linz, Austria, 2006, pp. 1223-1230.

[4] A. Kobsa, "User Modeling and User-Adapted Interaction", *CHI' 94*, Tutorial.

[5] J. Fink, A. Kobsa, "A Review and Analysis of Commercial User Modeling Servers for Personalization on the World Wide Web", *User Modeling and User-Adapted Interaction*, Vol. 10, Kluwer Academic, 2000, pp. 209-249.

[6] Y. S. Chen and G. D. Magoulas, "Adaptable and Adaptive Hypermedia Systems", IRM Press, 2005.

[7] C. Boyle and A. O. Encarnacion, "Metadoc : An adaptive hypertext reading system", User Modeling and User-Adapted Interaction, Kluwer Academic, Vol. 4(1), 1994, pp. 1–19.

[8] E. Rich, "Stereotypes and User Modeling", User Models in Dialog Systems, Springer, 1989, pp. 35-51.

[9] J. Fink, A. Kobsa and A. Nill, "Adaptable and Adaptive Information Provision for All Users, Including Disabled and Elderly People", New Review of Hypermedia and Multimedia, Vol. 4, 1998, pp. 163-188.

[10] S. S. Brown and P. Robinson, "Transformation frameworks and their relevance in universal design", Universal Access in the Information Society, Vol. 3.3-4, pp. 209-223, 2004.

[11] B. Encelle and N. Jessel, "Adapting presentation and interaction with XML documents to user preferences", Lecture Notes in Computer Science, Springer, vol. 3118, ICCHP'04: International Conference on Computers Helping People with Special Needs, Paris, France, 2004, p. 143-150.

[12] B. Encelle and N. Jessel, "Using the concept of user policies for improving HTML documents accessibility", IADIS Press, ICWI'04 Vol. 2, IADIS International Conference WWW/Internet 2004, Madrid, Espagne, 2004, pp. 835-839.

[13] B. Encelle and N. Jessel, "Personalization of information browsing user interface: evaluation of the user policies concepts", ICWI'06: IADIS International Conference WWW/Internet 2006, Murcia, Espagne, 2006.

[14] W3C, XML Path Language (XPath) http://www.w3.org/TR/xpath

[15] W3C, Synchronized Multimedia Integration Language (SMIL 2.0). http://www.w3.org/TR/smil20/

[16] Aho, A., Sethi R. and Ullman, J. Compilers, principles, techniques and tools. Addison-Wesley, 1986.

[17] JavaCC - Java Compiler Compiler [tm] (JavaCC [tm]) - The Java Parser Generator - https://javacc.dev.java.net/

[18] B. Encelle, "Accessibilité aux documents électroniques: personnalisation de la présentation et de l'interaction avec l'information.", Ph.D. Thesis, University Paul Sabatier, Toulouse, France, 2005.

[19] W3C, XSL Transformations (XSLT), http://www.w3.org/TR/xslt

[20] J. Coutaz,L. Nigay, D. Salber, A. Blandford, J. May and R. Young, "Four Easy Pieces for Assessing The Usability of Multimodal Interaction: The CARE properties", InterAct'95, Lillehammer, 1995, pp. 115-120.

[21] W. Meier, "eXist: An Open Source Native XML Database",In Revised Papers from the NODe 2002 Web and Database-Related Workshops on Web, Web-Services, and Database Systems, Springer-Verlag, 2003, pp. 169-183.

[22] W3C - Web Accessibility Initiative,http://www.w3.org/WAI

[23] I. Ramakrishnan and A. Stent and G. Yang, Hearsay: Enabling audio browsing on hypertext content, In Intl. World Wide Web Conf. (WWW), 2004, pp. 80-89.

[24] P. Brusilovsky, "Adaptive Hypermedia", User Modeling and User-Adapted Interaction, Vol. 11.1-2, Kluwer Academic, 2001, pp. 87-110.

[25] J. Palsberg and C. Barry Jay, "The Essence of the Visitor Pattern", Proc. 22nd IEEE Int. Computer Software and Applications Conf., COMPSAC, 1998, pp. 9-15.