How to build an adaptive web site: A Framework for User Adaptation

Daisuke Kanjo
National Institute of Information
and Communications Technology
Kyoto, Japan
kanjo@nict.go.jp

Yukiko Kawai
National Institute of Information
and Communications Technology
Kyoto, Japan
yukiko@nict.go.jp

Katsumi Tanaka
Kyoto University
Kyoto, Japan
tanaka@dl.kuis.kyoto-u.ac.jp

Abstract

We propose a framework for a user adaptation, called Adaptation Anywhere & Anytime(“A3”). User profiles for a user adaptation are automatically constructed and shared between web sites on A3. We also provide web-sites authors with a simple method for building an adaptive web site on A3 using XSLT.

1 Introduction

Today, we can get numerous informational resources from various web sites. So, web sites are required to adapt their output, for example, to select resources appropriate to individula user’s interests, needs, knowledge, etc. and present them. To provide a basis for selecting appropriate resources, web sites need to collect user profiles. In addition, they must be capable of using user profiles to select or present resources.

To make web sites adaptable to individual users, we propose a framework called Adaptation Anywhere & Anytime(“A3”). A3 provides web sites with methods for acquiring a user profile and selecting resources according to the user profile. With A3, web sites become adaptive to users’ needs. To simplify the implementation of an adaptive web site, a simple and easy method for implementation is needed. So, we decided to enhance an XSLT, which is a popular method for building a web site.

2 A3 Framework

Figure 1 shows the architecture of A3. A user accesses web sites built on A3 from various devices such as a PC, PDA, cellular phone, etc. Web sites adapt their output to the user based on his/her user profiles using the methods provided by A3. On A3, a user profile is represented by *user ontology*. A user ontology is defined as a classified tree written by RDF/S[Beckett, 2004; Brickly *et al.*, 2004] and OWL[Dean *et al.*, 2004],which is used to select suitable resources. User ontologies consist of resources and its categories. The resource is that have previously been presented to the user and that the user has acquired knowledge about. Each category has restrictions. The restriction is an attribute. The resources categorized into the category must have. Because each user acquires different knowledge about a resource, each user has his/her own user ontology.

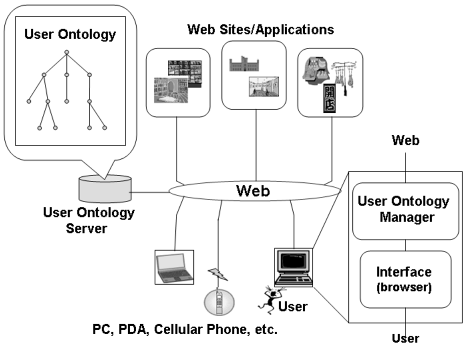

Figure 1: Overview of A3 framework

User ontologies are automatically constructed based on interactions between the user and the web site. For example, when a user enters data for a resource according to the requirements of the web site, it is reasonable to consider that the user has knowledge of that resource. On such situation, the resource is added to the appropriate category of the user ontology. If necessary, a new category is made or the ontology is reconstructed. In addition, user ontologies are shared by web sites and used as user profiles for selecting suitable resources.

Sharing user ontologies and automatic construction of ontologies resolve some problems of conventional adaptive systems, for example, differences in the adaptive precision between systems, or requiring users to answer questionnaire. However, there are another problems concerning users’ privacy and the consistency of user ontologies.

Consistency of user ontologies User ontologies are constructed by adding resources to an ontology. This requires node adding/deleting manipulations[1], such as appendChild(), removeChild(), etc. of DOM. However, if individual web site are free to construct user ontologies using these manipulations, inconsistency can occur in ontologies.

To maintain the consistency of user ontologies, web sites are not permitted to carry out node adding/deleting manipulations directly. They are permitted to determine when a resource should be added to a user ontology.

User’s privacy To select a resource using user ontologies,

[1] Resources and categories can be represented by nodes on user ontologies

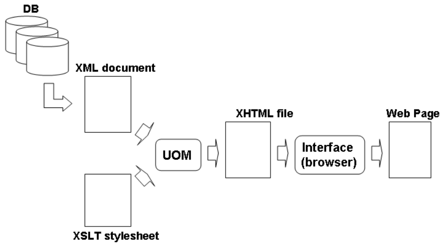

Figure 2: A3 resource selection Process

node referring manipulations such as getNodeName(), getParentNode(), etc. of DOM, are needed. These manipulations might invade a user privacy because they specify resources or categories in the user ontology. Therefore, to maintain user privacy, web sites are not permitted to carry out node referring manipulations.

To solve these problems, the manipulations described above are performed by the UOM.

UOM: User Ontology Manager The UOM located in the devices has two main functions: to construct user ontologies and select suitable resources.

When a user enters data of a resource, it can be considered that the user has knowledge about the resource. Then, a web site decides that a resource should be added to a user ontology, and informs it to the UOM. The UOM adds the resource to the user ontology.

When a web site want to present resources to a user, it sends all the candidate resources to the UOM. The UOM then calculates the weight of each resource. The weight of a resource in category C_n is given by $\sum_{k=1}^{n}(\frac{1}{2})^{k-1} * w_k * d_{tmp}$. w_k is the weight of the restriction of category C_k, which is between C_n and the root category (C_0). w_k is given by p/q. p is the number of resources in the user ontology with the attribute $Attr$ specified by the restriction of C_k. And q is the number of resources in the user ontology with the attributes $Attrs$. $Attrs$ are attributes that are common to all the candidate resources presented by the web system. d_{tmp} is a numerical value from 0 to 1. d_{tmp} decreases gradually according to the period since the resource was last categorized into C_k. Then, highest resources are selected.

The UOS is also a main component of the A3.

UOS: User Ontology Server The UOS stores user ontologies while users are not accessing a web site. When a user starts to access a web site on A3, the UOS sends the user's ontology to the UOM. The UOM uses it to select appropriate resource. Then, when the user finishes accessing the web site, the ontology is returned to the UOS for storage. Thus, the same user ontology is shared and used to select resources.

2.1 Implementation of web site on A3

To simplify implementation of a web site on A3, we decided to use XSLT. Figure 2 shows a schematic of the A3 process based on XSLT. The only task the web-site author has to do is to write an XML document and XSLT stylesheet. The XML document contains all the candidate resources that can be presented to the user. The XSLT stylesheet specifies which resources to selected. To specify those resources, an `a3:sort` element is used. This element work like an `xsl:sort` element, but it sorts resources accroding to the users ontology. The following is an example of part of a stylesheet.

```
<xsl:apply-templates>
     <a3:sort maxN="10" />
</xsl:apply-templates>
```

The `a3:sort` element is a child element of `xsl:apply-templates`. When the user accesses a web site, the XML document and XSLT stylesheet are sent to the UOM from the web site. Then, the UOM transforms the XML document into an XHTML file according to the XSLT stylesheet. If the UOM finds a `a3:sort` element in the XSTL stylesheet, it calculates the weight of the resources that match the `xsl:apply-template` element and selects the highest weighted resources. The selected resources are written in XHTML file, which is then transformed into a web page at the interface.

3 Related work

Personis[Kay *et al.*, 2002] is a server system for user modeling for an adaptive hypermedia system. To acquire a user profile, Personis explicitly requires users to answer some questions.

Research on user modeling is discussed by [Dolog and Nejdl, 2003] and research on ontology mapping or merging by [Noy and Musen, 2003]. These discussions focus on construction user models, mapping to different ontologies, or building tools to construct user models or ontologies. They do not consider how to use the constructed ontology.

4 Conclusion

We proposed a framework for a user adaptition, called Adaptation Anywhere & Anytime ("A3"). Its features is sharing user ontologies and automatic construction of ontologies. We also provide simple method for implementation of web site on A3 using XSLT.

References

[Beckett, 2004] Beckett,D., RDF/XML Syntax Specification, *http://www.w3c.org/TR/rdf-syntax-grammar/.*

[Brickly *et al.*, 2004] Brickley,D.,Guha,R.V., RDF Vocabulary Description Language 1.0:RDF Schema, *http://www.w3.org/TR/rdf-schema/*

[Dean *et al.*, 2004] Dean,M., Connolly,D., Harmelen,F., Hendler,J., Horrocks,I., McGuinness,D., Schneider,P., Stein,L., OWL Web Ontology Language Reference, *http://w3.org/TR/owl-ref/.*

[Dolog and Nejdl, 2003] Dolog,P.,Nejdl,W., Challenges and Benefits of the Semantic Web for User Modeling, *Adaptive Hypermedia and Adaptive Web-Based Systems*, 2003

[Kay *et al.*, 2002] Kay,J.,Kummerfeld,B.,Lauder,P., Personis: A Server for User Models *Proceedings of Adaptive Hypermedia and Adaptive Web-Based*, pages 203–212, 2002

[Noy and Musen, 2003] Noy,N.F.,Musen,M.A., The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983-1024, Dec. 2003.