A3 : Framework for User Adaptation using XSLT

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

Yukiko Kawai
National Institute of
Information and
Communincations Technology
Seikachyo, Kyoto, Japan
kanjo@nict.go.jp

Katsumi Tanaka
Kyoto University,
Department of Social
Informatics,
Yoshida Honmachi, Sakyo-ku,
Kyoto, Japan
tanaka@dl.kuis.kyoto-u.ac.jp

ABSTRACT

We propose a system called "Adaptation Anywhere & Anytime(A3)", which is a framework for making web sites/applications adaptable to user's needs or interests, and we describe the implement of a web site on A3 by using XSLT. Web sites/applications built on A3 construct user ontologies for each user automatically and share them between sites/applications. Each site/application uses the user ontology to select an appropriate resource for the user and to present such resources in a suitable form. And A3 offers the method for constructing the adaptable web sites using XSLT. The author of web sites can easily make their sites adaptable by using XSLT.

Categories and Subject Descriptors:

I.2.m[Artificial Intelligence]Miscellaneous

General Terms: Design

keywords: User Adaptation, Ontology, XSLT, Semantic Web

1. INTRODUCTION

Today we can get a lot of information from the web, but most of them are not what we are looking for. So a user adaptation technique is required to locate and present the information for which the user is looking. Generally, to achieve a user adaptation, it is necessary to collect user profiles. In previous adaptation systems, there are two major approaches for collecting user profiles: preprocessing as is done with a questionnaire, and automatic collection based on the interaction between the user and the system. However, preprocessing is very expensive for users. The cost of automatic collection is not higher than the cost of preprocessing. But when systems use the acquired information only for their own system, a difference of the accuracy of adaptation may occur. For example, the system which the user frequently uses can achieve an adaptation very well, but the system which the user rarely uses can not. To solve this problem, the user must repeat a similar procedure for each system. This is a burden to the user.

To solve these problems, we propose "Adaptation Anywhere & Anytime (A3)", which is a framework for making web sites/applications adaptable to user's needs or interests. The approach adopted by A3 is automatic construction and sharing of the user ontology. The user ontology is defined as a classified tree of web resources, and is written by RDF(S)[2, 3] and OWL[6] proposed for Semantic Web[1]. We assume that the user ontology represents the user's knowledge, and we use the user ontology as an user profile to achieve a user adaptation on A3. Sharing and automatic construction of the user ontology solves the problem of the cost and multiple
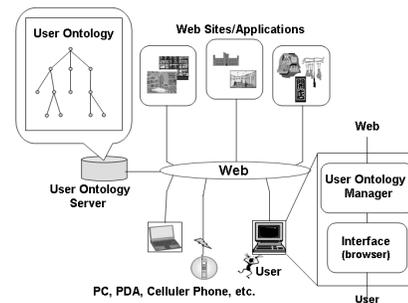
Figure 1: An Overview of A3 Framework

required similar procedure, which plagues the conventional system. That is, the automatic construction of the user ontology reduces the user's cost of answering the questionnaire. And by sharing the user ontology, the system can achieve an adaptation using the user ontology constructed by another systems, so user must not repeat a similar prcedure.

2. OVERVIEW OF A3 FRAMEWORK

Figure 1 shows an overview of the A3 framework. The user browses web sites or uses the web applications using a user interface(browser) on terminal devices such as PCs, PDAs, cellular phones, etc. Web sites/applications built on A3 use the user ontology to enable a user adaptation. While the user is browsing the site or using the application, a user ontology is automatically constructed based on the interaction between the user and the site/application. The constructed user ontology is shared by sites/applications.

In sharing a user ontology, each user must have one user ontology, not many ontologies. For this purpose, the user ontology server is connected to the web.

User Ontology Server (UOS) : The UOS saves the user ontology while the user is not using any web sites/applications. When the user starts to use site/application on A3, the UOS sends that user's ontology to the user ontology manager. There, the user ontology is used for achieving a user adaptation and is reconstructed based on the interaction between the user and the site/application. When the user finishes using the site/appl-ication, the reconstructed user ontology is returned to the UOS. Thus, same user ontology is shared between systems, so the user adaptation is achieved based on the same user ontology anytime and anywhere.

Furthermore, sharing the user ontology causes some problems

concerning user ontology consistency, user privacy, etc. For example, if each site/application constructs a user ontology using its own policy, each user ontology won't be consistent from site to site or application to application. To solve these problems, the user ontology manager performs all actions on the user ontology.

User Ontology Manager (UOM) : UOM has two main task: the construction of the user ontology, and achieving an user adaptation such as selecting the appropriate resource to the user's needs using the user ontology. To solve the problem which the sharing of the user ontology causes, the direct action such as the construction and the use of the user ontology is performed by UOM, is not performed by sites/applications. Each site/application can decide when and which action is done, but not how to do it. According to the decision made by the system, the UOM decides how to achieve each action and actually performs it.

3. IMPLEMENTATION OF WEB SITE

In easily and simply adapting the web site to the user, we decided to develop a method for implementation of the web site using XSLT[5]. In A3, XSLT is used to translate an original XML/XHTML file written by the author of web sites into XHT-ML file containing appropriate resources for the user. A3 offers some new XSLT elements for sorting of resources, construction of the user ontology and so on. These new elements are achieved using extension elements and extension functions of XSLT. In the following, it is assumed that the author of the web site wants to sort resources, for example, a book, according to the user's preference and present them on the web page. The example of XML/XHTML file is :

```
<?xml version="1.0"?>
 <?xml-stylesheet type="text/xml" href="book.xsl"?>
  <BOOKS>
      <BOOK>
          <AUTHOR>Asimov</AUTHOR>
          <TITLE>I, Robot</TITLE>
          ...
      </BOOK>
      <BOOK>
          <AUTHOR>Clarke</AUTHOR>
          <TITLE>2001: A Space Odyssey</TITLE>
          ...
      </BOOK>
      ...
  </BOOKS>
```

The XML/XHTML file contains the data of resources. Generally, the example of such resources are news stories on news sites, merchandise of e-shopping sites and so on. In this case, resources are books. "book.xsl" (written by italic) presents XSLT stylesheet of this XML file. It is :

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://wwww3org/1999/XSL/Transform">
    <xsl:output method="html">
    ...
    <xsl:template match="/BOOKS">
        <HTML>
            ...
            <xsl:apply-templates>

            <a3:sort/>

            </xsl:apply-templates>
            ...
        </HTML>
    </xsl:template>

    <xsl:template match="BOOK">
        <xsl:apply-templates select="TITLE"/>
    </xsl:template match="BOOK">
</xsl:stylesheet>
```

This XSLT stylesheet contains an <a3:sort> element (written by bold). This element is the additional element by A3 and works like an <xsl:sort> element. While an <xsl:sort> element sorts resources matching to <xsl:template> element according to one of their attributes of them, an <a3:sort> element sorts that resources using the user ontology. When the user starts to use the site having these files, the XML/XHTML file and the XSLT stylesheet are sent to the UOM. The UOM sorts the resources matching <xsl:templates

match="BOOK"> element (i.e. books) based on the user ontology, and creates a new XHTML file. In this new file, books appear in order of the result of sorting. The book which is the most interesting to the user will appear in first, the one which is of least interesting will appear in last. Then, new XHTML file is sent to the browser and translated into a web page.

The only task the author of web sites has to perform is to write the XSLT stylesheet and XML/XHTML file. If the author of web site provides these files, resources matching <xsl:template> element are sorted and then presented in this way if it is the first time the user uses that site. And sorting is done even if the author doesn't know how to sort resources.

The sorting of the resources is done described below. The user ontology consists of category and resources. Every resource is categorized into one category. Each category has a restriction. The restriction is an attribute-value pair to which the resource categorized to that category must satisfy. The UOM calculates the weight w of the attribute-value pair. The weight w is given by p/q. For example, p is the number of resources that were categorized to the category to which the resource being sorted is categorized. And q is the number of resources that match to the <xsl:template> element and were categorized to some category of the user ontology. And the weight of category C is given by : $\sum_{k=1}^{n} \left(\frac{1}{2}\right)^{k-1} * w_k$. n is the number of categories between category C and the root category. w_k is the weight of the attribute-value pair which each category between category C and the root has. The weight of the root is w_0 and the weight of the Category C is w_n. The weight of the resource is the weight of a category the resource is categorized in. According to the weight of each resource, resources are sorted.

To construct a user ontology, the resource are added to the user ontology, when the user can be regarded as she/he had acquired information about the resource. The web system requires the UOM to add the resource to the user ontology. According to this requirement, the UOM adds a resource using the following three procedures:(1) categorizing the resource into an existing category, (2) making a new category and categorizing the resource into the new category, or (3) deleting unneccessary cetegories.

4. CONCLUSION AND FUTURE WORK

We proposed A3, which is a framework for user adaptation. In this framework, the automatic construction of the user ontology and the sharing of the user ontology are enabled in achieving user adaptation. Furthermore, A3 has a method of ensuring the easy implementation of the web site using XSLT. So A3 is very useful for the web site authors wishing to make their sites adaptable. But, there is still work that must be done. The conventional methods of A3 are limited, so we need to enhance them. Moreover, it is necessary to actually build sites/applications on A3 and verify its functionality, and to improve the adaptation accuracy and so on.

5. REFERENCES

[1] Berners-Lee,T.,Hendler,J.,Lassile,O., The Semantic Web, Scientific American(2001)
[2] Beckett,D., RDF/SML Syntax Specfication, http://www.w3c.org/TR/rdf-syntax-grammar/
[3] Brickley,D.,Guha,R.V., RDF Vocabulay Description Language 1.0:RDF Schema, http://www.w3.org/TR/rdf-schema/
[4] Chafee,J.,Gauch,S., Personal Ontologies for Web Navigation, In Proc. of CIKM'00(2000)
[5] Clark,J., XSL Transformations(XSLT) Version1.0, http://www.w3.org/TR/xslt
[6] Dean,M.,Connolly,D.,Harmelen,F.,Hendler,J.,Horrocks,I., McGuinness,D.,Schneider,P.,Stein,L.,OWL Web Ontology Language 1.0 Reference,http://w3.org/TR/owl-ref/
[7] Pretschner,A.,Gauch,S., Ontology Based Personalized Search, In Proc. 11th Intl. Conf. on Tools with Artificial Intelligence,(1999)