

AN EXTENSIBLE TECHNIQUE FOR CONTENT ADAPTATION IN WEB-BASED INFORMATION SYSTEMS

Roberto De Virgilio and Riccardo Torlone

Dipartimento di Informatica e Automazione

Università Roma Tre

Roma, Italy

{devirgilio,torlone}@dia.uniroma3.it

Abstract: Adaptive Web information systems need to exploit information about the context of the client in order to deliver, in an appropriate way, relevant information. In this paper, we present a general approach to this problem that is able to handle heterogeneous context information and different coordinates of adaptation. The approach is based on a general notion of profile that can be used to represent a variety of contexts at different level of details. Client profiles, possibly expressed in different formats, are wrapped and translated into such a general representation. The analysis of profiles drives the generation of an interface configuration. A configuration specifies, at the various layers of a Web based Information System (content, navigation and presentation), how to build a response that meets the requirements of adaptation of the profile. We describe architecture and functionality of a prototype implementing such adaptation methodology and illustrate practical examples of use of the prototype.

Keywords: Adaptive Information Systems, Context awareness, Data Intensive Web Sites, Heterogeneous profiles

1. INTRODUCTION

An ever increasing number of mobile devices, such as PDAs and next generation phones, can be deployed today to provide *everywhere* and *any time* access to the Web. This scenario presents a number of challenges, especially when it comes to data-intensive applications, mainly because these devices offer limited computing capabilities. It follows that a novel and fundamental requirement of modern Web based Information Systems is the ability to adapt and personalize content delivery according to the *context* of the client (a human being or an application). There

is no accepted definition of context, but the term is a usually adopted to indicate “a set of attributes that characterizes the capabilities of the access mechanism, the preferences of the user and other aspects of the context into which a Web page is to be delivered” (W3C Device Independence Working Group, 2003). These may include the access device, the network QoS, the user preferences, the location, and so on.

Many approaches have been proposed to the problem of the design and the implementation of an adaptive Web based information system (Bickmore et al, 1999; Ceri et al, 2000; Ceri et al, 2003; Kießling, 2002; Leubner and Kie, 2002; Gu and Helal, 2004; Pastor et al, 2003; Schwabe et al, 1996). However, they are often specific solutions, suited only for predefined adaptation requirements (usually device characteristics and user preferences) and hardly reusable for adding new adaptation functionality to existing systems. In these approaches a relevant problem that has received little attention is the large heterogeneity of formats used to express context information: text files in ad-hoc format, HTTP headers, XML files over specific DTD’s, RDF (Lassila and Swick, 1999; Brickley and Guha, 2003), CC/PP (Klyne et al, 2004).

In our study, we focus our attention on the large category of *data intensive* Web Information Systems, which mainly provide a Web access to large amounts of structured data, and address the problem of providing a general solution to the problem of content adaptation that can be used for different and possibly independent requirements of adaptation.

To this end, we first present a very general notion of *profile* that can be used to represent a large variety of aspects of a context, at different level of details. Each profile is associated with a *configuration* that specifies, in an abstract way, how the Web pages that compose the response to deliver to the user have to be generated. This is done by taking into account both the specific user request and his/her context. Profiles can be compared and configurations can be merged: these features are used for reusing configurations specified for “compatible” profiles and for integrating different requirements of adaptation.

The activities of the various components of the architecture are coordinated by a general methodology for content adaptation, based on the generation and management of configurations. In order to experiment the effectiveness of our approach, we have designed and developed a prototype implementing the proposed methodology. The tool is based on a very general architecture for an adaptive system that can be easily extended to meet new requirements of adaptation, not fixed a-priori. The tool makes use of a caching technique that take advantage of a repository of configuration implementations to make the adaptation process more efficient. A involved component of the tool is the profile interpret

that takes as input incoming profiles expressed in a variety of formats and converts them in a common format that represents the context in terms of our general model of profile.

The paper is structured as follows. In Section 2 we present a general model for adaptive Web-based information systems. In Section 3, we present the general methodology of adaptation, by illustrating the basic concepts of *profiles* and *configuration* and showing their use in a practical example. In Section 4 we first present a general architecture and then illustrate a practical implementation of the system based on this architecture. Finally, in Section 5 we draw some conclusions.

2. A MODEL FOR ADAPTIVE IS

In this section we present a simple but general model that we use as a reference for our approach. It is based on two main “logical” constructs: the profile and the configuration.

2.1 Profiles

A *profile* is a description of an autonomous aspect of the context in which the Web site is accessed (and that should influence the presentation of its contents). Examples of profiles are the user, the device, the location, and so on. A *dimension* is property that characterizes a profile. Each dimension can be conveniently described by means of a set of attributes that associates actual values with a dimension. For instance, a profile for a client device can be represented by means of the hardware, software, and browser dimensions. In turn, the hardware dimension can be described by means of attributes like CPU, memory, and display. Within a dimension, it is useful to introduce a partial order between attributes at different level of details. As an example, in Figure 1 are reported possible dimensions for the profile of a device. In the HARDWARE dimension, the HD spec attribute fully describes the hardware capabilities of the device, whereas the Display attribute just represents the display dimensions of the device. Possible profiles over these dimensions are reported in the bottom of the same figure.

A *context* is simply a collection of profiles. Note that this notion of context is very general and is therefore suited to model almost all context formalisms proposed in the literature and adopted in practical systems.

In our model, different profiles over the same dimensions can be compared making use of a subsumption relationship \triangleleft . Intuitively, this relationships compares the level of detail of the profiles: given two profiles P_1 and P_2 , we say that P_1 subsumes P_2 ($P_1 \triangleleft P_2$) if P_2 includes all the attributes of P_1 at the same or at a coarser level of detail, according to

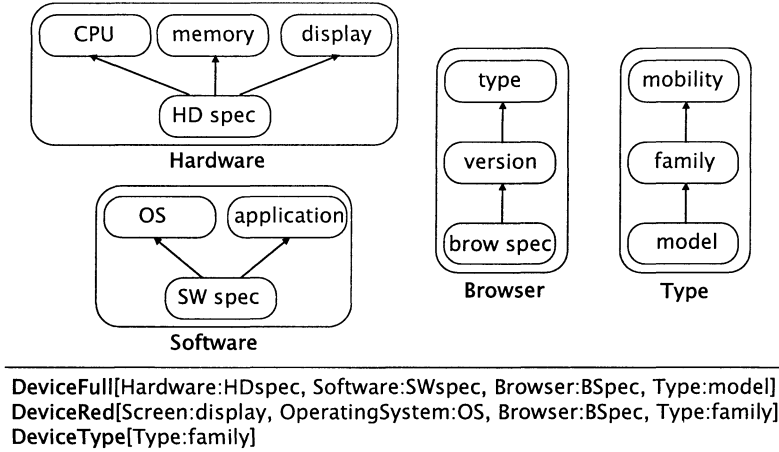


Figure 1. Dimensions for a device and possible profiles over them

the hierarchy defined on the attributes of the dimensions. We assume that \triangleleft has a unique top element called the *generic* profile. As an example, the profile `DeviceType` in Figure 1 subsumes the profile `DeviceRed` that, in turn, subsumes the profile `DeviceFull`.

2.2 Configurations

Let us now turn our attention to an adaptive Web-based Information System (WIS) that is able to modify and personalize delivery of contents and services according to different profiles. As we have said in the introduction, we focus our attention on the large category of data intensive WIS, which mainly provide a Web access to large amounts of structured data. In such applications, it is useful to consider separately its three main components: the content (that is, the data to publish), the presentation (that is, the layout of the pages) and the navigation (that is, the hypertext structure of the Web site). It is agreed (Fiala et al, 2003; Fiala et al, 2004; Frasinca et al, 2004; Vdovjak et al, 2003) that an adaptation process should operate on all these components: selecting the most appropriate content (e.g., according to user interests), building an adequate layout for the web pages (e.g., according to the layout capabilities of the client device) and organizing the hypertext structure of the web interface (e.g., decomposing large contents in linked pages, when the band of the communication channel is limited). This approach has been

According to this observation, we introduce the abstract notion of (*interface*) *configuration* as a triple $C = (q, h, p)$ where:

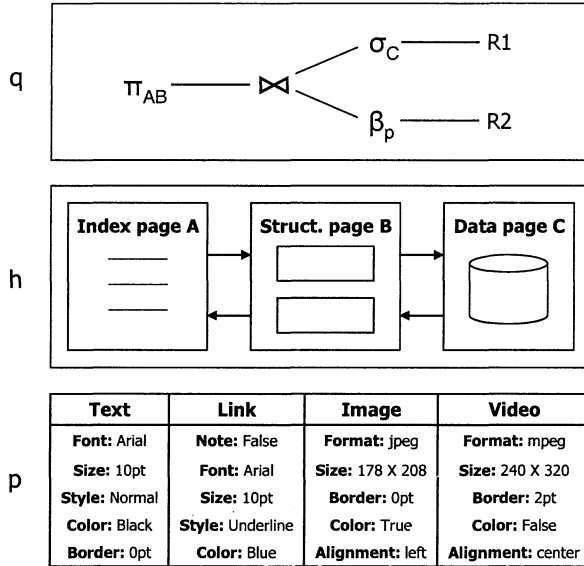


Figure 2. A configuration $C = (q, h, p)$

- q is a query over the underlying database expressed in standard relational algebra augmented with a special operator, called *best*, that can be used to specify queries based on qualitative preferences (Chomicki, 2003; Torlone and Ciaccia, 2002);
- h is an abstract hypertext definition expressed in WebML (Ceri et al, 2003), a conceptual model for Web application which allows us to describe the structure of Web pages in a tight and concise way, by abstracting their logical features;
- p is presentation specification, expressed in terms of an original notion of *logical style sheet* that will be described in more detail below.

A Web page is composed by a set of *Web objects*, each of which is classified according to a predefined set of *Web Object Types (WOTs)*. Possible WOTs are *text*, *image*, *video*, *form* and so on. Each WOT τ is associated with a set of *presentation attributes*: they identify possible styles (e.g. font, color, spacing, position) that can be specified for τ . Given a set $W = \{w_1, \dots, w_n\}$ of WOTs, a *logical style sheet* (or simply a *lss*) for W is a set of possible values for w_1, \dots, w_n , respectively. Clearly, we can have different lss's for the same set of WOT's.

As an example, a graphical representation of a configuration $C = (q, h, p)$ is reported in Figure 2. In this figure, q has been represented by a tree, h by a site view (Ceri et al, 2003), and p by a table.

It is important to note that a configuration is indeed a logical notion that can be represented and implemented in several ways and with different syntaxes. This property guarantees the generality of the approach with respect to actual languages and tools used to implement the adaptive application. For instance, we can implement a configuration using SQL at the content level, XHTML at the navigation level and a set of CSS files at the presentation level.

We now introduce a composition operation over configurations, denoted by \oplus , that will be used in the adaptation process to merge different configurations. Given a pair configurations $C_1(q_1, h_1, p_1)$ and $C_2(q_2, h_2, p_2)$, $C_1 \oplus C_2$ is a configuration $C(q, h, p)$ defined as follows:

- $q = q_1 \circ q_2$, that is, q is obtained as the composition of q_1 followed by q_2 ;
- h is obtained by merging h_1 and h_2 : if some conflict arises, the choices of h_1 are preferred to those of h_2 ; and
- $p(w_i) = p_1(w_i)$ if w_i is a WOT occurring in p_1 and $p(w_i) = p_2(w_i)$ otherwise (that is, if w_i is a WOT occurring only in p_2).

Note that the \oplus operation is indeed a *prioritized* composition since if two configurations present conflicts, then the choices done in the configuration on the left hand side of \oplus are preferred to the choices done in the other configuration.

Finally, a configuration can be associated with a profile to specify, in an abstract way, an adaptation suitable for the profile. In this case we say that the configuration *matches* the profiles. We do not give a precise definition of matching since, in general, this relationship is very difficult to establish in an automatic way. Therefore, we assume, at this stage, that the matching between profiles and configurations is predefined (e.g., specified by the designer of the application). We are currently investigating under which circumstances a match between a profile and a configuration can be derived automatically.

3. A GENERAL METHODOLOGY OF ADAPTATION

3.1 The Adaptation Process

Our process of adaptation is based on the notions of profiles and configurations presented above. It relies on an initial set of configurations

C that capture the criteria of adaptations for a basic set of profiles. The only requirement is that **C** contains at least one configuration that match the generic profiles of a context. Clearly, this set needs to be refined and enriched during the life cycle of the adaptive system.

The methodology of adaptation can be summarized as follows:

- 1 First, the context of the client is captured and represented in terms of a set of profiles, one for each coordinate of adaptation. Each profile is expressed in the model presented in the previous section. As usual, some aspects of the context can be provided explicitly by the client (e.g., device capabilities), others can be specified implicitly (e.g., user preferences can be derived by the analysis of his/her navigation).
- 2 For each profile P of the context:
 - (a) we select in **C** a configuration that matches with P ;
 - (b) if there is no configuration in **C** that matches P , we select in **C** a configuration that matches a profile P' that subsumes P .
- 3 The set of configurations generated in the previous step are merged into a unique configuration C making use of the \oplus operator, on the basis of a predefined order of precedence on the various coordinates (e.g., the device capabilities take precedence over the preferences of the users).
- 4 The configuration C can be further refined to meet the requirements of a specific request done by the user (e.g., an additional data selection);
- 5 The final configuration C is translated into a corresponding set of *adaptation statements* that implement the configuration in the actual languages for the systems adopted at the various levels. Adaptation statements may correspond to: SQL statements at the content layer, XHTML or JSP statements at the navigation layer, and CCS style sheets at the presentation layer.
- 6 The adaptation statements are executed by the underlying systems and the final response is generated.

3.2 A Practical Example

Let us consider a data intensive Web site that publishes news items taken from different newspapers and assume that we intend to make it adaptive to different device characteristics and user preferences.

In Figure 3 is reported the whole process of adaptation when the site is accessed by a user having a preference of news and sport information with his/her cellular phone having a black and white display of limited size and without graphical capabilities.

First of all, the context of the client is captured and expressed in terms of a pair profiles P_d and P_u describing, respectively, the device and the user preferences. According to the technique presented in the previous section, two configurations are selected, one for each profile. The pair of configurations are then merged and we obtain in this way a final configuration representing, in an abstract way, the following adaptations.

- At the content level, the adaptation consists in a query that combines a projection introduced by P_d , which eliminates attributes that the device cannot display (e.g., pictures), and a selection introduced by P_u , which retrieves only news items preferred by the user, as indicated in the top of Figure 3.
- At the navigation level, the adaptation consists in the definition of an hypertext scheme that combines a distribution of contents into linked pages of limited size, to meet the requirements of P_d , and the separation of summaries and actual news, to meet the requirements of P_u ; the scheme and the corresponding pages are reported in the middle of Figure 3.
- At the presentation level, the adaptation consists in the specification of black and white colors for P_d and of a font resize, as specified by the user in P_u ; the logical style sheet we obtain is reported in the bottom of Figure 3 together with the corresponding output.

Note that the “logical” adaptations specified in the final configuration needs to be translated into actual specifications that can be then executed over the various levels to generate the desired interface.

4. IMPLEMENTATION

4.1 An Architecture of Reference

A general architecture of a system able to meet the requirements of adaptation described in the Introduction and to implement the proposed methodology of adaptation is reported in Figure 4.

This includes:

- a Request Interpreter (RI), able to translate a specific user request (a page or a specific object) into a query over the underlying data,

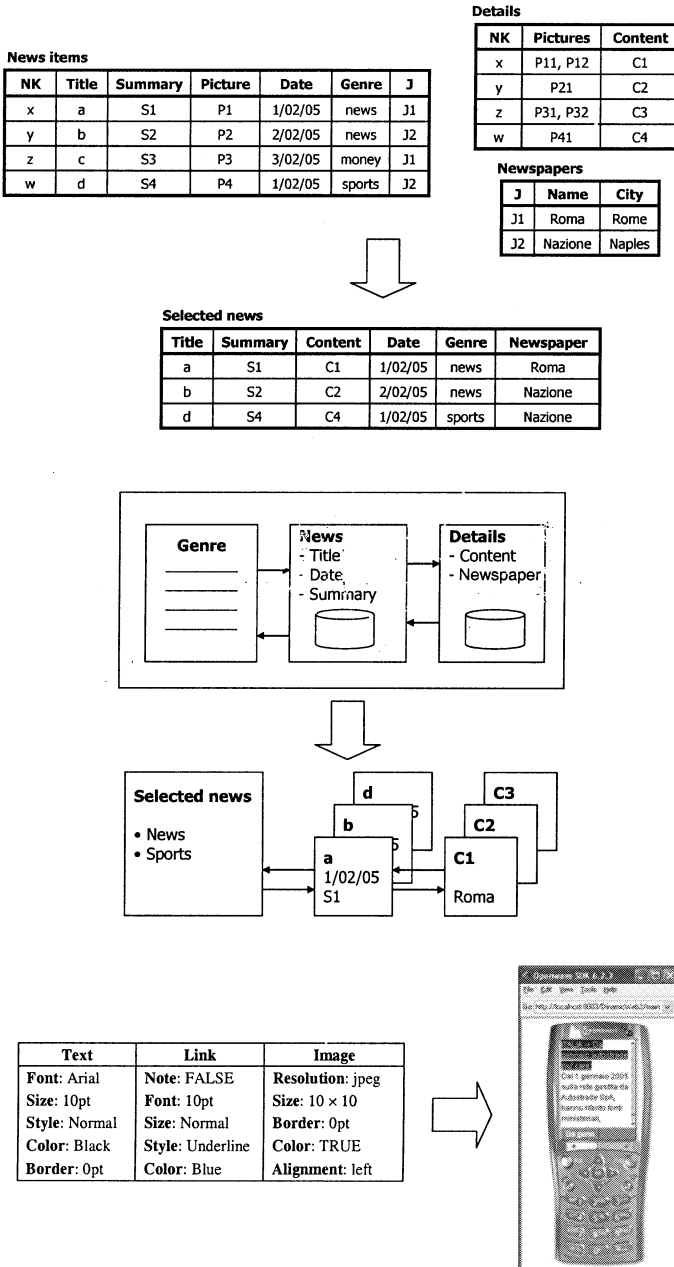


Figure 3. Adaptation at the various levels of a data intensive Web site

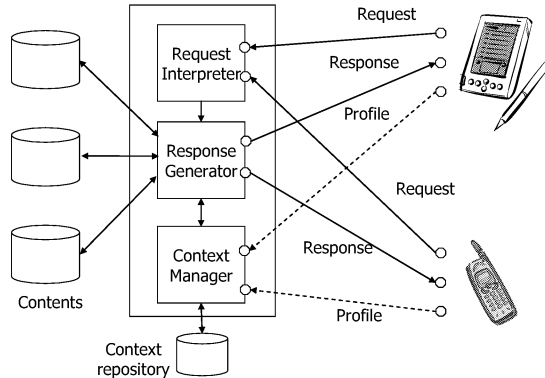


Figure 4. A general architecture of reference

- an Response Generator (RM), able to generate all the components of a response to deliver over the Web (content, structure and layout) that satisfies the given request and is appropriate for the client profile.
- a Context Manager (CM), able to get and manage a description of the client characteristics (the profiles) and support the Response Generator in the execution of its task.

Clearly, the fundamental component of this architecture is the Context Manager that should be able to:

- 1 (dynamically) capture and classify (possibly heterogeneous) incoming profiles of clients making use of a local repository of profiles,
- 2 coordinate the various (and possibly conflicting) requirements of adaptation for a given profile,
- 3 send to the Response Generator some *adaptation specifications* at all the levels (content, navigation and presentation) of the response.

To guarantee the flexibility of the overall system, this component should be extensible, in the sense that the various activities should be carried out for different types of profiles and according to orthogonal dimensions of adaptation, possibly not fixed in advance.

In Figure 5 it is reported a possible architecture for the Context Manager that can meet these requirements. The basic component of this

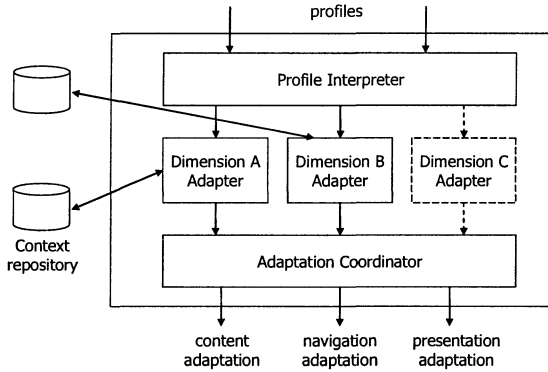


Figure 5. An Extensible Context Manager

module is the Profile Interpreter, which should be able to get and identify possibly heterogeneous profiles (e.g., CC/PP, XML, HTTP headers) and translate them into a uniform representation expressed in the model proposed in Section 1. Such profile representations are taken as input by a series of modules, one for each dimension of adaptation (e.g., the device characteristics, the user preferences, the location, etc.). The main task of these modules is to generate a uniform set of adaptation specifications, expressed in terms of a configurations, that satisfy the specific requirements of one dimension. This work can be supported by a special data repository in which predefined or previously generated profiles are stored together with their corresponding configurations. Since each module can generate different and possibly conflicting configurations, a coordination based on the \oplus operator is performed to provide an integrated configuration that take into account the various adaptation requirements and can be effectively sent to the RG module. The Adaptation Coordinator is devoted to the execution of this task.

It is important to note that, due to the uniformity of representations and techniques used by the various adaptation modules, this scheme can be extended in a natural way: a new adaptation module can be easily added to satisfy the requirements of adaptation of a previously unpredicted coordinate.

A Response Generator that can match the other components of our architecture is composed by three modules (figure 6), one for each levels of the response to deliver over the Web. The first module combines the first component of the configuration provided by the Context Manager and generates from it a query to be executed by a Query Processor (possibly external to the system). The second module operates over the

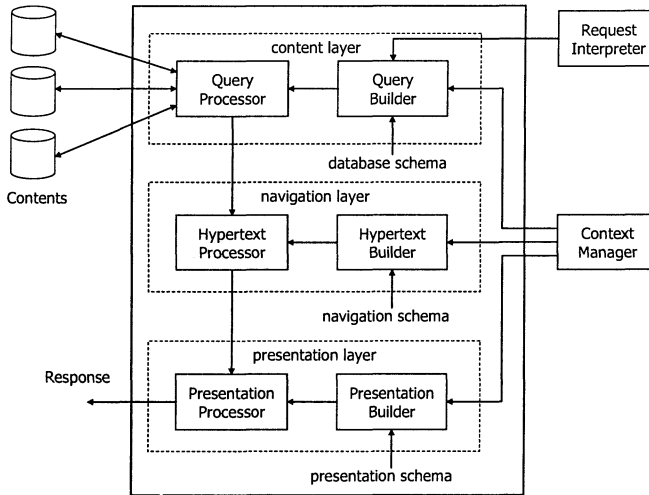


Figure 6. A Response Generator

navigation scheme of the Web site (e.g., by splitting pages or adding links) to satisfy the requirements of adaptation specified by the second component of the selected configuration. Finally, the last module is in charge of implementing the last component of the configuration at hand with an appropriate style sheet.

4.2 FAWIS

We have designed and developed the first release of a system called FAWIS (Flexible Adaptation of Web-based Information Systems) to test the our approach. The architecture of FAWIS is based on the general architecture reported in Figure 4 and also includes a Session Manager that is able to infer the interests of the user on a set of Web pages according the chronology of his/her navigation. The system relies on an XML database of contents and the selection of data is done by performing XML queries expressed in XQuery.

Currently, two modules of adaptation have been implemented, devoted to the management of the device capabilities and the user preferences, respectively. The repository for the former coordinate has been built by using the WURFL database (<http://wurfl.sourceforge.net/>), which includes information about capabilities and features of a large set of wireless devices currently available on the market. *Qualitative* preferences (Kießling, 2002; Torlone and Ciaccia, 2002) are used to represent the user profiles. The selection of contents based on qualitative



Figure 7. Three different results generated by FAWIS

preferences makes use of a special optimization technique that we have studied elsewhere (Torlone and Ciaccia, 2002).

In Figure 7 is reported the final results of the adaptation process performed by FAWIS, for the same information content, according to three different devices and user preferences.

In order to improve the efficiency of the system, we have implemented an optimization strategy based on the reuse of adaptation statements (see Section 3) and a cache of results. Statements defined for implementing a certain configuration are stored by the Response Generator in a special repository. When a new configuration C is generated by the Context Manager, the system verifies whether a suitable set of statements implementing C is already present in the repository. In this case, the system avoids the need for generating them again and efficiently executes the statements by taking advantage of a cache of already computed results.

An involved component of FAWIS is the profile interpreter that is currently able to interpret incoming profiles expressed in several languages including HTTP headers, UAProf (UApof, 2001), CC/PP, and plain XML. These are translated into an homogeneous internal representation that refers to our profile model and is given as input to the various adaptation modules. We have equipped the Profile Interpreter with a graphical interface that allows the user to define new formats of profiles and specify mappings between the various formats and the general representation. These mappings are used to perform the translation of the incoming profiles.

We have tested our system to experiment the effectiveness and the efficiency of the approach illustrated in this paper in several practical cases.

On the server side, we used an IBM computer xSeries 225, equipped with a Xeon2 2.8Ghz processor, a 4 GB RAM, and a 120 GB HDD SCSI. The Web site have been accessed by three different types of devices: a mid-range desktop, several PDAs with different capabilities, and some cellular phones.

In general, the results obtained so far are promising. The effectiveness of the approach is supported by the capability demonstrated by the system to generate satisfactory results for different contexts, even not fixed in advance. This is due to the ability of the approach to select configurations for “compatible” context, as described in Section 3. Also, it turned out that the adaptation process can be executed always in a reasonable amount of time and that the response time rapidly improves when the caching of adaptation statements becomes effective.

5. CONCLUSION AND FURTHER WORK

In this paper, we have presented a general approach to the problem of content delivery adaptation of Web information. The approach is based on a general notion of profile that can be used to represent a variety of contexts at different level of details. Each profile is associated with a configuration that specifies, in abstract terms, how the response for a given request has to be delivered over the Web, by taking into account the requirements of adaptation for the profile. We have presented a general methodology for content adaptation based on the generation and management of configurations. We have also described and tested a prototype implementing the proposed approach.

From a conceptual point of view, we are currently investigating in more depth the notions of profile and configuration, in order to improve their generality and usability. From a practical point of view we are extending the features of the tool, by enhancing the profile interpretation capabilities and by adding new adaptation modules.

References

- T. Bickmore, A. Girgensohn, and J. Sullivan. Web page filtering and reauthoring for mobile users. *Computer Journal* 42(6) (1999), 534–546.
- D. Brickley and R. Guha. *RDF vocabulary Description Language 1.0: RDF Schema*. W3C Working Draft, 10 October 2003.
- S. Ceri, P. Fraternali, and A. Bongio. Web modeling language (webml): a modeling language for designing web sites. In *9th International Conference on the World Wide Web (WWW9) Amsterdam*, 2000.
- S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai, and M. Matera. *Designing Data-Intensive Web Applications*. Morgan Kaufmann, 2003.

- J. Chomicki. Preference formulas in relational queries. *ACM Trans. Database Syst.* 28(4): 427–466, 2003.
- Z. Fiala, M. Hinz, K. Meissner, and F. Wehner. A component-based approach for adaptive dynamic web documents. *Journal of Web Engineering, Rinton Press* 2:058–073, 2003.
- Z. Fiala, F. Frasinicar, M. Hinz, G.J. Houben, P. Barna, K. Meißner. Engineering the presentation layer of adaptable web information systems. In *International Conference on Web Engineering (ICWE 2004) Glasgow, Scotland*, (July 28-30 2004).
- F. Frasinicar, P. Barna, G.J. Houben, and Z. Fiala. Adaptation and reuse in designing web information systems. In *International Conference on Information Technology, Track on Modern Web and Grid Systems, (IEEE Compute Society 2004) Glasgow, Scotland* (387–391).
- W. Kießling. Foundations of preferences in database systems. In *(VLDB 2002) pp. 311-322* (2002).
- G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. Butler, and L. Tran. *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies*. W3C Working Draft, 2004.
- W. Gu and A. S. Helal. An XML Based Solution to Delivering Adaptive Web Content for Mobile Clients. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04), San Jose, California, July 25–29* (2004).
- S. Gupta, G. Kaiser,, D. Neistadt, and P. Grimm. Dom-based content extraction of html documents. In *Twelfth International Conference on the World Wide Web (WWW2003), Budapest, Hungary* (2003).
- O. Lassila and R. Swick. *Resource description framework (rdf) model and syntax specification*. W3C Working Draft, 22 February 1999.
- A. Leubner and W. Kießling. Personalized keyword search with partial-order preferences. In (SBBD 2002: 181-193), 2002.
- Open Mobile Alliance WAP Forum 2001. *Wireless Application Group: User Agent Profile Specification*.
- O. Pastor, J. Fons, and V. Pelechano. A method to develop web applications from web-oriented conceptual models. In *International Workshop on Web Oriented Software Technology (IWWOST) ((2003) 144–173)*.
- D. Schwabe, G. Rossi, and S.D.J. Barbarosa. Systematic hypermedia application design with oohdm. In *Hypertext '96, The Seventh ACM Conference on Hypertext ACM Washington DC* (1996), 116–128.
- R. Torlone and P. Ciaccia. Which are my Preferred Items? In *2nd Int. Workshop on Recommendation and Personalization in e-Commerce, Malaga, Spain*, pag. 1–9, 2002.
- R. Vdovjak, F. Frasinicar, G.J. Houben, and P. Barna. Engineering semantic web information systems in hera. *Journal of Web Engineering, Rinton Press* 2 (2003), 003–026.
- M. Wagner, W. Balke, W. Kießling. An xml-based multimedia middleware for mobile online auctions. In *(ICEIS(2)2001: 934-944)* (2001).
- W3C Working Group on Device Independence. Device Independence Principles. *Internet document: <http://www.w3.org/TR/di-princ/>*, 2003.