

A General Methodology for Context-Aware Data Access

Roberto De Virgilio and Riccardo Torlone
Dipartimento di Informatica e Automazione
Università di Roma Tre
{devirgilio,torlone}@dia.uniroma3.it

ABSTRACT

In this paper, we present an extensible approach to the adaptation of Web information delivery according to different and possibly heterogeneous contexts. 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 information has to be delivered by taking into account the requirements of adaptation for the profile. We present a general and extensible methodology for content adaptation based on the generation and management of configurations. The genericity of the approach guarantees that different types of contexts and orthogonal requirements of adaptation, possibly not fixed in advance, can be taken into account in the adaptation process. We also describe architecture and functionality of a prototype implementing the proposed methodology. We finally provide a number of experimental results of the system to support the effectiveness and efficiency of the approach.

Categories and Subject Descriptors

H.2.8 [Information Systems]: Database Management—
Database applications

General Terms

Management, Design

Keywords

Adaptive Information Systems, Context awareness, Data Intensive Web Sites

1. INTRODUCTION

The Web is today the universal interface to information and the increasing number of devices able to provide *everywhere and any time* access to the Web is a clear witness of this trend. As it often happens however, the rapid and uncontrolled spreading of these devices has led in many cases

a number of previously uncharted and rather involved challenges.

A novel and fundamental requirement in this scenario is the ability of the system 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 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” [19]. 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 (from its birth and creation until usage) [1, 4, 5, 6, 8, 13, 10, 18, 11, 15, 16]. These approach provide a solution for the whole development process: design, implementation, configuration and deployment. However, they are often specific solutions, suited only for predefined coordinates of adaptation (usually device characteristics and user preferences) and hardly reusable for adding new adaptation capabilities to existing systems.

In this paper, 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 and scalable solution to the problem of content adaptation that can be used for different and possibly independent requirements of adaptation.

To this end, we first propose a very general architecture for an adaptive system that can be easily extended to meet new requirements of adaptation, not fixed a-priori. We then 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. Configurations can be compared and merged and this is used 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 architecture and methodology. This system makes use of a matching technique between

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiDE'05, June 12, 2005, Baltimore, Maryland, USA.
Copyright 2005 ACM 1-59593-088-4/05/0006 ...\$5.00.

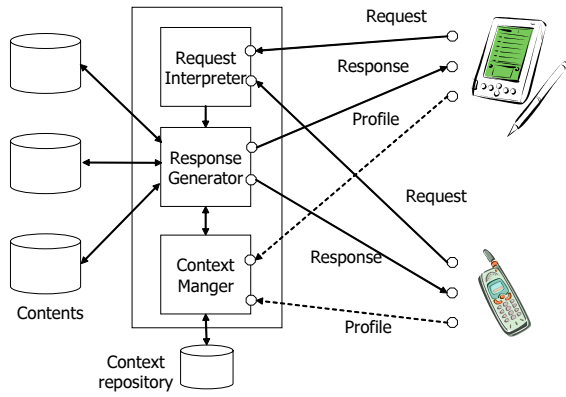


Figure 1: A general architecture of reference

profiles and a repository of configurations to make the adaptation process more efficient. We illustrate a number of experiments that have been conducted with this system and present quantitative results to support the feasibility of the overall approach and the efficiency of the optimization technique.

The paper is structured as follows. In Section 2 we describe the architecture of reference. In Section 3, we present the general methodology of adaptation, by illustrating the basic concepts of *profiles* and *configuration* and showing how they are used in the system. In Section 4 we present a practical implementation of the system and some experimental results of use. Finally, in Section 5 we draw some conclusions.

2. AN EXTENSIBLE APPROACH

2.1 An architecture of reference

A Web-based Information System (WIS) is adaptive if it is able to modify and personalize delivery of contents and services according to the context of the client. 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). 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).

It turns out that a general architecture of a system able to meet these requirements is the one reported in Figure 1. 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,
- an Response Generator (RM), able to generate all the components of a response to deliver over the Web (con-

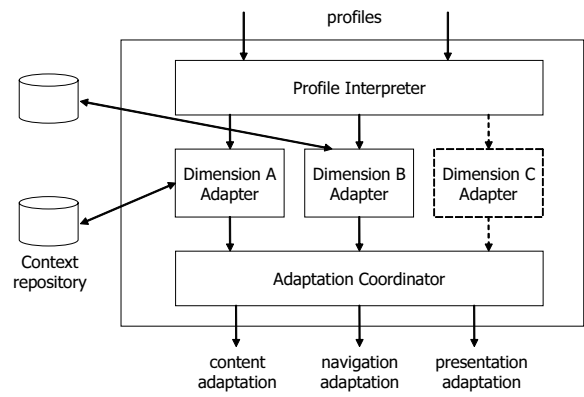


Figure 2: An Extensible Context Manager

tent, 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 (usually called profile) 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 2 it is reported a possible architecture for the Context Manager that can meet these requirements. The basic component of this 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. 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, to be sent to the Response Generator, 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 specifications. Since each module can generate different and possibly conflicting specifications, a coordination is needed to provide an integrated set of specifications that take into account the various adaptation requirements and can be effectively sent to the RG

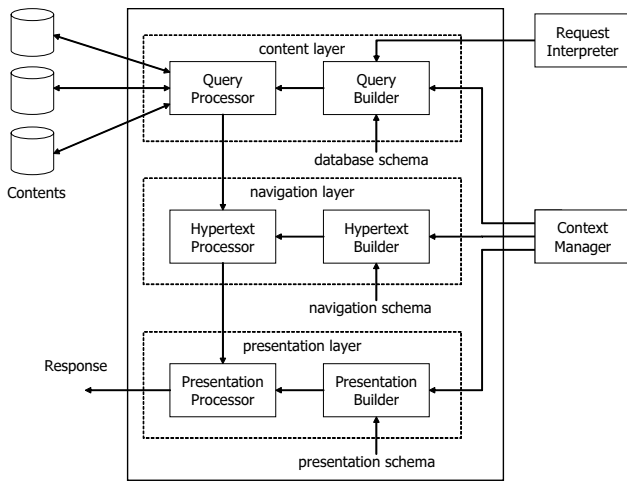


Figure 3: A Response Generator

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 with the other components of our architecture is composed by three modules (figure 3), one for each levels of the response to deliver over the Web. The first module combines the query returned by the Request Interpreter with the adaptation specifications provided by the Context Manager and generates a query to be executed by a Query Processor (possibly external to the system). The second module operates over the navigation scheme of the Web site (e.g., by splitting pages or adding links) to satisfy the requirements of adaptation, as specified by the CM. Finally, the last module is in charge of taking the specifications of adaptation related to the presentation and implementing them with an appropriate style sheet.

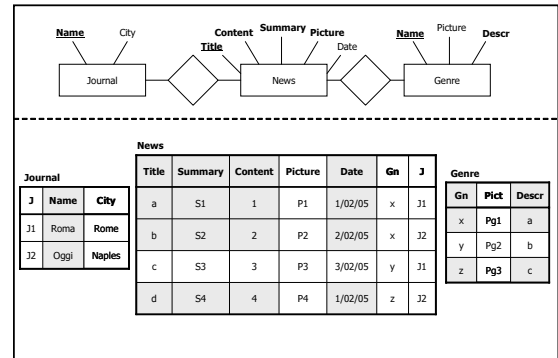
In the rest of the paper, we will present in more detail the methodology that leads the various activities of the Context Manager and illustrate an implementation of this approach for two specific adaptation dimensions. For simplicity, we will focus our attention on the adaptation at the presentation layer.

2.2 A practical example

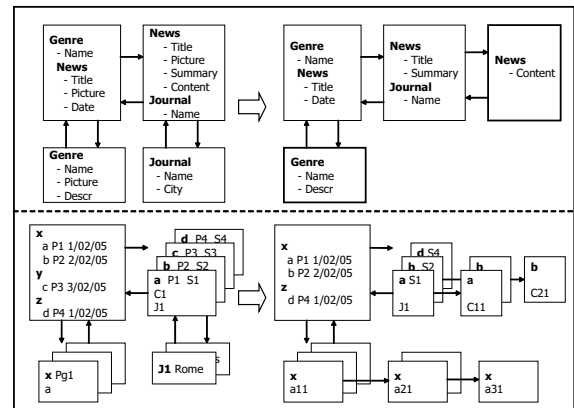
Let us consider a data intensive Web site that publishes news taken from different newspapers and assume that we intend to make it adaptive to different device characteristics and user preferences.

In Figure 4 is reported the whole process of adaptation when the site is accessed by a user having a preference of summaries for genre x with his/her cellular phone with a black and white display of limited size and without graphical capabilities.

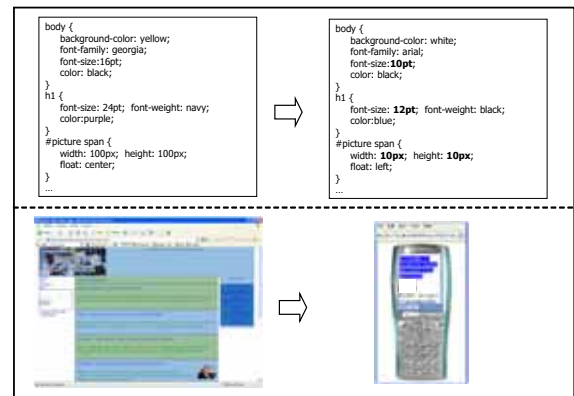
In Figure 4 is reported the whole process of adaptation when the site is accessed by a user having a preference of summaries for genre x with his/her cellular phone with a black and white display of limited size and without graphical capabilities.



(a)



(b)



(c)

Figure 4: Adaptation at the various levels of a data intensive Web site

(UA), take as input these profiles and each of them generates three sets of adaptation specifications, one for each layer of the Web site. The specification are defined at schema level and operates at instance level.

- **Content:** the DA performs a vertical filtering by eliminating some attribute that the device cannot display (e.g., pictures), whereas the CM performs a horizontal filtering by selecting only news preferred by the user, as indicated in Figure 4.a.
- **Navigation:** the DA redefines the hypertext structure of the site by introducing links and decomposing large pages to take into account the display limitation of the device, whereas the UA isolates summaries from news by introducing a link; the resulting scheme is reported in Figure 4.b.
- **Presentation:** the DA imposes only black and white colors, whereas the UA resizes fonts as specified by the user; the generated style sheet is reported in Figure 4.c.

The Adaptation Coordinator of the CM combines the various specifications and sends three integrated sets of specifications to the Response Generator, which actually executes the adaptation over the various levels. The final results at instance level are also reported in Figure 4.

3. THE ADAPTATION METHODOLOGY

3.1 Profiles and Configurations

Let us start with a simple but general model that we use as a reference for our approach.

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. For example, 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. A context is a collection of profiles.

A subsumption relationship \triangleleft can be defined over profiles. Intuitively, given two profiles P_1 and P_2 , if $P_1 \triangleleft P_2$ then P_2 is more detailed than P_1 since it includes the attributes of P_1 at the same or at coarser level of detail and possibly more. We assume that \triangleleft introduces a partial order on profiles that has a unique top element called the *generic* profile. Examples of profiles for access devices, ordered according to \triangleleft , are reported in Figure 5.

Note that this notion of profile is rather general and is therefore suited to model a variety of aspects.

Let us now turn our attention to contents. We consider a Web page as a *box* in which different *Web objects* can be inserted. Each object is classified according to a predefined set of *Web Object Types (WOTs)*. Possible WOTs are *text*, *image*, *video*, *form* and so on. Each level of the site has its own attributes. At the presentation level, each Web Object Type 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 . Each attribute has a

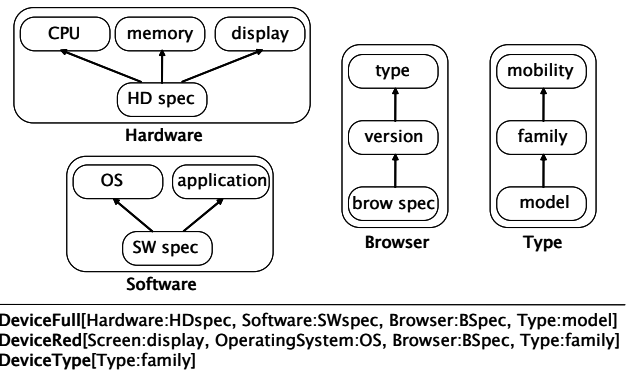


Figure 5: A collection of profiles

Text	Link	Image	Video
Font: Arial	Note: FALSE	Resolution: jpeg	Resolution: mpeg
Size: 10pt	Font: 10pt	Size: 176 × 208	Size: 240 × 320
Style: Normal	Size: Normal	Border: 0pt	Border: 2pt
Color: Black	Style: Underline	Color: TRUE	Color: FALSE
Border: 0pt	Color: Blue	Alignment: left	Alignment: center

Figure 6: A set of Web Object Types and a configuration over it

domain associate with it, that is, a set of possible values for the attribute.

Similarly to profiles, we assume that a subtype relationship, denoted by \triangleleft , is defined over Web Object Types. The top element of the hierarchy is the *generic object type* and has associated a set of basic attributes. As usual, given two WOTs w_1 and w_2 , if $w_1 \triangleleft w_2$ then all the attributes of w_2 are inherited by w_1 . For instance, we can define an *audio* WOT as a subtype of a *multimedia* WOT. The relation \triangleleft can be extended to a set of WOTs in a natural way: given two sets of WOT W_1 and W_2 , $W_1 \triangleleft W_2$ if for each WOT $w_1^i \in W_1$ there is a WOT $w_2^i \in W_2$ such that $w_1^i \triangleleft w_2^i$.

Given a set $W = \{w_1, \dots, w_n\}$ of WOTs, a *configuration* for W , denoted by $C(W)$, is a set of instances $C(W) = \{c(w_1), \dots, c(w_n)\}$ of w_1, \dots, w_n , respectively. Clearly, we can define different configurations for the same set of WOTs. Note that configuration is an abstract entity that can be implemented in several ways and with different syntaxes. A CSS style sheet or an XSL file are examples of configuration implementations. A simple example of a configuration over a set of WOTs is reported in Figure 6

Two configurations $C_1(W_1)$ and $C_2(W_2)$ can be compared according to a subsumption relationship denoted by \sqsubseteq and defined as follows: $C_1(W_1) \sqsubseteq C_2(W_2)$ if $W_1 \triangleleft W_2$ and, for each WOT w_i in W_1 , $c_1(w_i) = c_2(w_i)$.

Moreover, we introduce a composition operation over two configurations $C_1(W_1)$ and $C_2(W_2)$, denoted by \oplus and defined as follows: $C_1(W_1) \oplus C_2(W_2)$ is a configuration C over $W_1 \cup W_2$ such that $c(w_i) = c_1(w_i)$ if $w_i \in W_1$ and $c(w_i) = c_2(w_i)$ otherwise (that is, if w_i is a WOT occurring only in W_2).

Note that the \oplus operation is indeed a *prioritized* composition since if two configurations present different values over the same WOT, then the value for the configuration on the

left hand side of \oplus is preferred to the value for the other configuration.

A configuration can be associated with a profile to describe, in an abstract way, an adaptation suitable for the profile. In this case we say that the configuration *matches* the profiles.

3.2 The adaptation process

The process of content adaptation can be summarized as follows:

1. The Context Manager captures the context of the client and represents it as a set of profiles, one for each coordinate of adaptation; in general, some aspects of the context can be provided explicitly by the client, others (e.g., user preferences) can be derived implicitly (e.g., by the analysis of user navigation). Each profile is sent to the corresponding adapter module of the CM.
2. Each adapter module CA for the coordinate C receives a profile P of C and, taking advantage of a dedicated repository RC of configurations, performs the following tasks:
 - (a) it searches in RC for a configuration that matches with P ,
 - (b) if the search fails, it identifies a profile P' that subsumes P and search in RC for a configuration that matches with P' , and
 - (c) it sends the selected configuration to the Adaptation Coordinator.
3. The Adaptation Coordinator merges the configurations received using the \oplus operator and sends an integrated set of configurations to the Response Generator.
4. The Response Generator takes as input the configurations and generates from them a corresponding set of *adaptation statements* that implement the specifications at the various levels. Adaptation statements correspond to: queries at the content layer, hypertext manipulations at the navigation layer, and style sheets at the presentation layer.
5. The adaptation statements are executed by suitable processors, possibly external to the system itself.

4. A PRACTICAL IMPLEMENTATION

4.1 System features

We have designed and developed the first release of a adaptive system to test the effectiveness of our approach. The architecture of the tool is based on the general architecture described in Section 2 and includes also 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 on them. The system relies on an XML database of contents and the selection of data is done by performing XML queries expressed in XQuery.

The profile interpreter is currently able to interpret incoming profiles expressed in several languages including HTTP headers, UAProf [14], CC/PP, and plain XML. These are translated into an homogeneous internal representation that



Figure 7: Three different results generated by the systems

is given as input to an extensible set of adaptation modules. Currently, two modules of adaptation have been implemented, which are 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 (Wireless Universal Resource File) configuration database (<http://wurfl.sourceforge.net/index>). This repository includes information about capabilities and features of a large set of wireless devices currently available on the market. *Qualitative* preferences [8] are used to represent the user profiles. The selection of contents based on qualitative preferences makes use of a special technique that we have studied elsewhere [17].

In Figure 7 is reported the final results of the adaptation process performed by the system, 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 3.2) and a cache of results. Statements defined for implementing an adaptation specification are stored by the Response Generator in a special repository. When a new adaptation specification is generated by the Context Manager, the system verifies whether a suitable set of statements for it is already present in the repository. In this case, the system avoids the need for generating it again and efficiently executes the statements by taking advantage of a cache of already computed results.

4.2 Experimental results

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.

The results of our experiments are reported in Figure 8.

- Figure 8.a reports the average response time (vertical axis) for each device type (horizontal axis),
- Figure 8.b reports the response time (vertical axis)

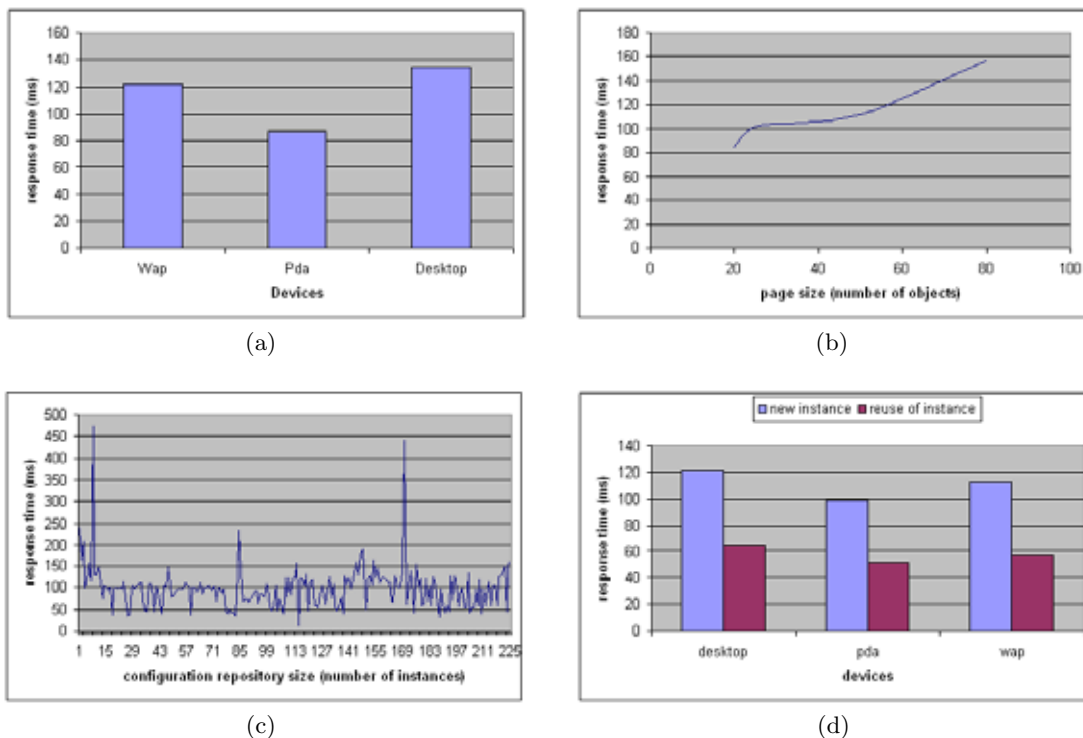


Figure 8: The experimental results

with respect to the size of the pages included in the response (horizontal axis);

- Figure 8.c reports the response time (vertical axis) with respect to the size of the repository of configurations (horizontal axis);
- Figure 8.d reports the average response time (vertical axis) with respect to each device type (horizontal axis), both in absence and in presence of the reuse technique described above.

In general, it turns out that the results obtained so far are very promising. They confirm that the adaptation process can be executed always in a reasonable amount of time and that the overall approach exhibits a good degree of scalability.

5. CONCLUSIONS AND FUTURE 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.

6. REFERENCES

- [1] T. Bickmore, A. Girgensohn, and J. Sullivan. Web page filtering and reauthoring for mobile users. *Computer Journal* 42(6) (1999), 534–546. In (SBBD 2002: 181-193), 2002.
- [2] P. Brusilovsky. Adaptive Hypermedia. *User Modeling and User-Adapted Interaction 11: 87–110* (2001), Kluwer Academic Publishers, Netherlands.
- [3] Sven Buchholz, Thomas Hamann, and Gerald Hbsch. Comprehensive Structured Context Profiles (CSCP): Design and Experiences. *Workshop on Context Modeling and Reasoning - CoMoRea'04*, (2004).
- [4] 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).
- [5] 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 (2003), 058–073.
- [6] Z. Fiala, F. Frasinca, 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).
- [7] F. Frasinca, 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).
- [8] W. Kießling. Foundations of preferences in database systems. In *(VLDB 2002) pp. 311-322* (2002).
- [9] 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.
- [10] 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).
- [11] 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).
- [12] O. Lassila and R. Swick. *Resource description framework (rdf) model and syntax specification*. W3C Working Draft, 22 February 1999.
- [13] A. Leubner and W. Kießling. Personalized keyword search with partial-order preferences.
- [14] Open Mobile Alliance WAP Forum 2001. *Wireless Application Group: User Agent Profile Specification*.
- [15] 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 (IWWOOST) ((2003) 144–173)*.
- [16] 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.
- [17] 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.
- [18] R. Vdovjak, F. Fransincar, G.J. Houben, and P. Barna. Engineering semantic web information systems in hera. *Journal of Web Engineering, Rinton Press 2* (2003), 003–026.
- [19] W3C Working Group on Device Independence. Device Independence Principles. *Internet document: <http://www.w3.org/TR/di-princ/>*, 2003.