World Wide Web
DOI 10.1007/s11280-007-0020-2

Rule-based Adaptation of Web Information Systems

Roberto De Virgilio · Riccardo Torlone · Geert-Jan Houben

Received: 18 April 2006 / Revised: 5 October 2006 /
Accepted: 8 January 2007

Abstract Mobile devices provide a variety of ways to access information resources available on the Web and a high level of adaptability to different aspects of the context (such as the device capabilities, the network QoS, the user preferences, and the location) is strongly required in this scenario. In this paper, we present a rule-based approach supporting the automatic adaptation of content delivery in Web Information Systems. The approach relies on the general notions of profile and configuration. The former is used to model a variety of context characteristics in a uniform way. The latter describes, in abstract terms, how to build the various levels of a suitable Web interface (content, navigation and presentation). We propose an original notion of adaptation rule that can be used to specify, in a declarative way, how to build a configuration that satisfies the requirements of adaptation for a profile. The evaluation process defined for these rules supports: (1) the handling of many separately specified adaptation requirements according to different aspects of the context, possibly not fixed in advance, and (2) their integration into one coherent recipe for adaptation. We also describe the architecture and functionality of a prototype implementing the proposed approach and illustrate experimental results supporting its flexibility and efficiency.

A preliminary version of this paper appears in the *Proceedings of Seventh International Conference on Mobile Data Management (MDM'06) Nara, Japan, May 10–12, 2006.*

R. De Virgilio · R. Torlone (✉)
Università Roma Tre, Rome, Italy
e-mail: torlone@dia.uniroma3.it

R. De Virgilio
e-mail: devirgilio@dia.uniroma3.it

G.-J. Houben
Vrije Universiteit Brussel, Brussels, Belgium
e-mail: Geert-Jan.Houben@vub.ac.be

Keywords mobile devices · web information systems · adaptive applications · context awareness · data intensive web sites

1 Introduction

The number and the spread of mobile devices able to provide access to the Web *anywhere* and *anytime* are increasing day by day. These devices include not only cellular phones, PDAs, and terminals for disabled people, but also new kinds of devices, possibly embedded into objects such as household appliances or vehicle dashboards. The characteristics of the various devices are so different that the delivering of information and services on the Web involves not only presentational aspects, as one might initially think, but also structural and navigational aspects. As an example consider a cellular phone: its limited computing capabilities require that information be filtered and organized as a collection of atomic units whose dimensions depend closely on specific features of the device. Considering the device capabilities is an example of a functionality that can be generalized as taking into account aspects characterizing the *context* in which Web information is to be delivered [31]. These aspects may include preferences or capabilities of the user, the QoS (Quality-of-Service) of the network, the location, the time, and so on. It follows that a modern context-aware, adaptive system for Web information should guarantee a high level of flexibility in terms of: (1) responsiveness to highly changing requirements of adaptation, and (2) suitable actions to be undertaken to meet these requirements.

In this paper we consider Web Information Systems (WIS) as the data-intensive Web applications than can benefit enormously from content-awareness in the delivery of the information. Approaches and methodologies for the design and development of WIS have been proposed in the last years and many have chosen a model-based approach [1, 6, 9, 14, 18, 21, 25–28, 30]. Many of the approaches and methodologies in this class address the issue of adaptation, i.e. they offer the designer possibilities to configure the particular context in which the information is delivered, e.g. [8, 15, 16]. However, the majority of the approaches can be viewed as specific solutions that are suited for a particular class of predefined adaptation requirements: most of them consider device characteristics and user preferences and allow the designer to describe the effect of the characteristics and preferences on the navigation and presentation. [20] describes how different interpretations of what is relevant regarding user and context leads even to different solution directions for adaptation and context-awareness. All of this makes it not exactly trivial when a new adaptation functionality needs to be added, since the adaptation design facilities need to be extended. In other words, in these approaches it is hard to take into account new aspects of the context and modify the adaptation accordingly.

With these limitations for extension and modification in mind, we propose in this paper a unifying methodology for the automatic adaptation of content delivery in Web Information Systems. That approach relies on two basic notions: the *profile* and the *configuration*. The profile is used to model a variety of contexts and their characteristics in a uniform way. The configuration describes, in abstract terms, how to build the various levels of a Web application (content, navigation and presentation).

The automatic adaptation of content delivery is achieved by means of an original notion of production rule that allows us to specify *declaratively* how to build a configuration satisfying the adaptation requirements for a given profile. The technique of evaluating rules supports partial matchings between contexts and rule heads. This guarantees that a large variety of contexts, possibly not fixed in advance, can be taken into account in the adaptation process. Moreover, given the diversity of design concerns and adaptation perspectives, it is a common problem in practice how to combine separate (aspect-specific) adaptation specifications into one global design of the application and its adaptation. For this reason the approach includes a method to combine configurations generated by different rules and to solve possible conflicts between them.

From a practical point of view, we illustrate the design of a prototype implementing the proposed methodology and propose an optimization strategy based on a rule clustering technique. We also present experimental results supporting the flexibility and efficiency of the system we have developed.

The rest of the paper is organized as follows. Section 2 introduces the scenario of reference for our investigation. In Section 3, we illustrate the basic notions of profile and configuration. In Section 4, we present the rule-based technique for the automatic generation of configurations and, in Section 5, we describe a practical implementation of the approach. Section 6 discusses the related work. Finally, in Section 7 we draw some conclusions and sketch future work.

2 The reference scenario

As we have stated in the Introduction, we focus our attention on the large category of *data-intensive* Web applications or Web information systems, which mainly provide Web access to large amounts of structured data. It is widely accepted that in the design of such applications it is useful to consider its three main components separately: the content, the navigation, and the presentation. As a consequence, a process of designing adaptation should reflect on all these components by selecting the most appropriate content (e.g., according to user interests), organizing the hypertext structure of the Web interface (e.g., by decomposing large contents into linked pages, whenever the communication channel's bandwidth is limited), and building an adequate layout for the Web pages (e.g., according to layout capabilities of the client's device).

As a running example for this paper to illustrate the approach, let us consider a data-intensive Web application that publishes news items taken from different newspapers. Assume that, at a given instant, the context is identified by a user who has a preference for recent summaries of sport events and is connected to a network without a guaranteed quality of service with his/her mobile phone having a black-and-white display of limited size and without graphical capabilities.

The first problem to face is how to gather and represent (model) such information about the context generally, given the large heterogeneity of formats used to express context data such as text files in ad-hoc format, HTTP headers, XML files over specific DTDs, RDF, CC/PP and their dialects.

Once the context has been captured, the adaptation process should identify and combine all the requirements coming from the different (and usually independent) aspects of the context. In our running example, some example requirements we could give at the various levels can be roughly summarized as follows:

– *Content*: select only sport summaries (user requirement), filter out all pictures (device requirement);
– *Navigation*: decompose large content chunks into a hyper-structure of linked pages (device requirement), keep the structure of linked pages compact and their number low (network requirement);
– *Presentation*: transform colors to grays, resize all text (device requirements).

Note that in some cases, conflicts between requirements can arise (in the example at hand, this happens at the navigation level) and the adaptation design should be able to detect and solve these situations. When a consistent set of requirements has been finally identified, the adaptation process should carry out suitable actions to meet these requirements along the various levels of the Web information system.

In Figure 1, we illustrate some possible adaptations for our example:

– At the content level only relevant news items have been selected,
– At the navigation level, the hypertext organization has been restructured by distributing the contents over a limited number of linked pages, and
– At the presentation level, the formatting instructions have been suitably rewritten for the device at hand.

An important point that needs to be taken into account is that this scenario is very dynamic: other users with new preferences can be enabled to access the system, possibly unpredicted types of access devices can be used, alternative network connections can be made available, and further aspects of the context (like the location or other environmental factors) want to be incorporated. Note that also the technology used to implement the adaptation can change and this would have an impact on the overall adaptive Web application.

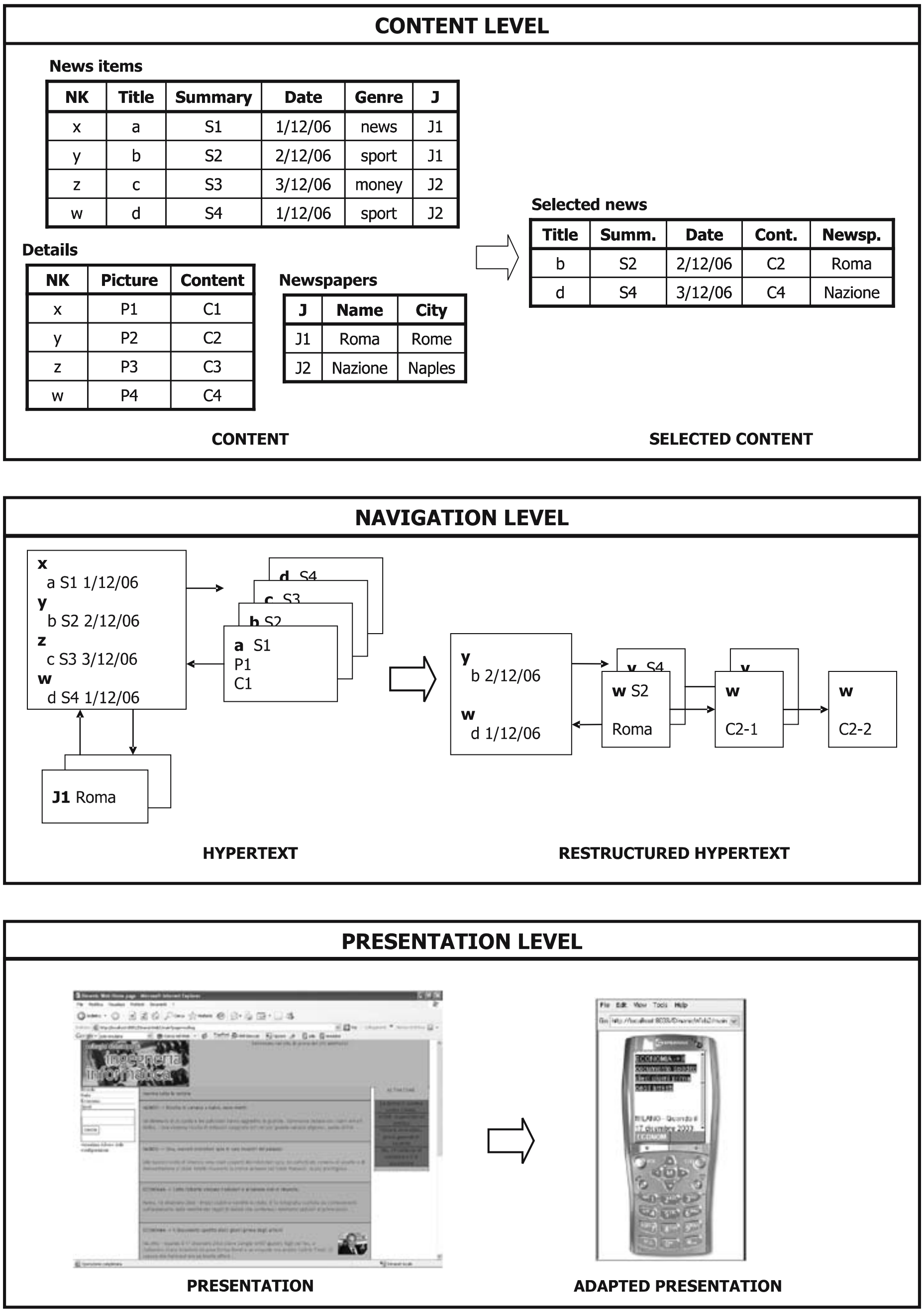
The rest of the paper is devoted to the presentation of conceptual and practical issues related to the definition of a general adaptation methodology that aims to satisfy these needs of generality and flexibility.

3 Profiles and configurations

In this section we present two basic notions on which our approach is based: the generic profile and the abstract configuration.

3.1 Generic profiles

In our approach, a *generic profile* is a description of an autonomous aspect of the context in which the Web site is accessed and that should influence the structuring and presentation of its contents. Examples of profiles are descriptions of the user, the device, the location, and so on. A *dimension* is a property that characterizes a profile. Each dimension is described by means of a set of *attributes*. For instance, a profile for a client device can be represented by means of the hardware, software, and browser dimensions. The hardware dimension can be described by means of attributes like CPU, memory, and display. Attributes can be *simple* or *composite*. A simple attribute



News items

NK	Title	Summary	Date	Genre	J
x	a	S1	1/12/06	news	J1
y	b	S2	2/12/06	sport	J1
z	c	S3	3/12/06	money	J2
w	d	S4	1/12/06	sport	J2

Details

NK	Picture	Content
x	P1	C1
y	P2	C2
z	P3	C3
w	P4	C4

Newspapers

J	Name	City
J1	Roma	Rome
J2	Nazione	Naples

Selected news

Title	Summ.	Date	Cont.	Newsp.
b	S2	2/12/06	C2	Roma
d	S4	3/12/06	C4	Nazione

Figure 1 Adaptation in a data-intensive Web information system.

has a domain of values associated with it, whereas a composite attribute has a set of (simple or composite) attributes associated with it. For example, the display attribute can be composed by the simple attributes width and height.

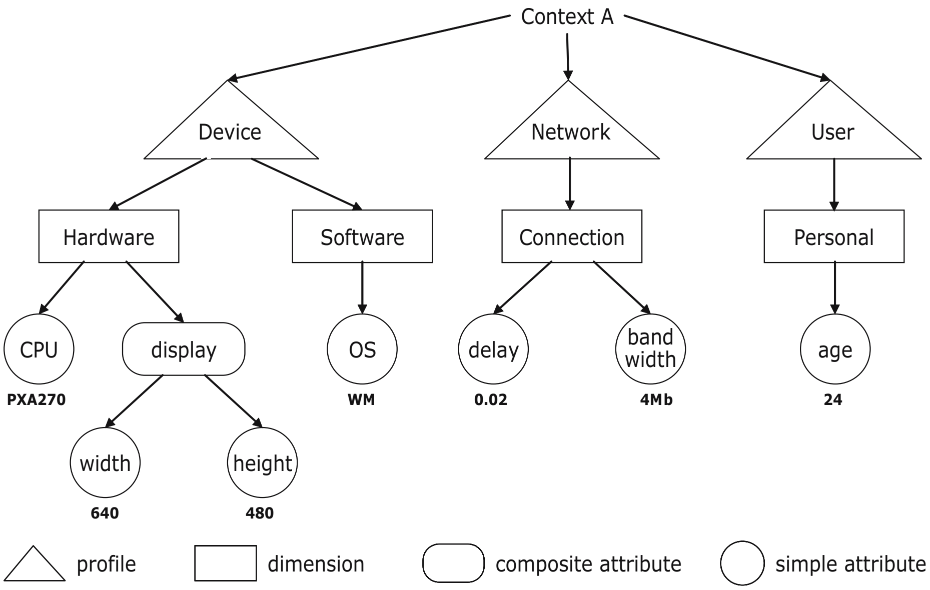


Figure 2 An example of context.

More precisely, let us fix a vocabulary $\mathbf{V} = \{\mathbf{D}, \mathbf{A}\}$ where $\mathbf{D}$ is a set of *dimension names* and $\mathbf{A}$ is a set of (simple and composite) *attributes names*. We denote by $D(A_1, \ldots, A_n)$ a *dimension definition*, where $D \in \mathbf{D}$ and $A_i \in \mathbf{A}$, for $1 \leq i \leq n$. If A_i is a simple attribute, then it is associated with a set of values called *domain*, otherwise, it has associated with a set $A_{i_1}, \ldots, A_{i_k}$ of (sub-)attributes.

Definition 1 (Profile and context) A (generic) *profile* P over a set of dimension definitions $D_1(X_1), \ldots, D_n(X_n)$ is function that associates with each simple attribute of every dimension a value taken from its domain. A *context* is a collection of profiles.

As an example, Figure 2 reports a graphical representation of a context composed by profiles for the device, the network and the user.

Note that our notion of context is very general and is therefore suited to model all the formalisms for representing context information proposed in the literature and adopted in practical systems. In fact, we have recently proposed a technique for the management of context information and the translation of context descriptions between heterogeneous formalisms and vocabularies [12]. This is achieved by using our model as an intermediate representation level.

A fundamental aspect in this framework is that different profiles can be compared making use of a subsumption relationship $\lhd$. Intuitively, given two profiles P_1 and P_2, if $P_1 \lhd P_2$ then P_2 is more detailed than P_1, since it includes all the components of P_1 at the same or at greater level of detail.

More precisely, we first say that an attribute A of a profile P *is covered* by an attribute A of a profile P' if either they are simple and $P(A) = P'(A)$, or they are composite and for each sub-attribute A_i of A in P there is a sub-attribute A_j of A

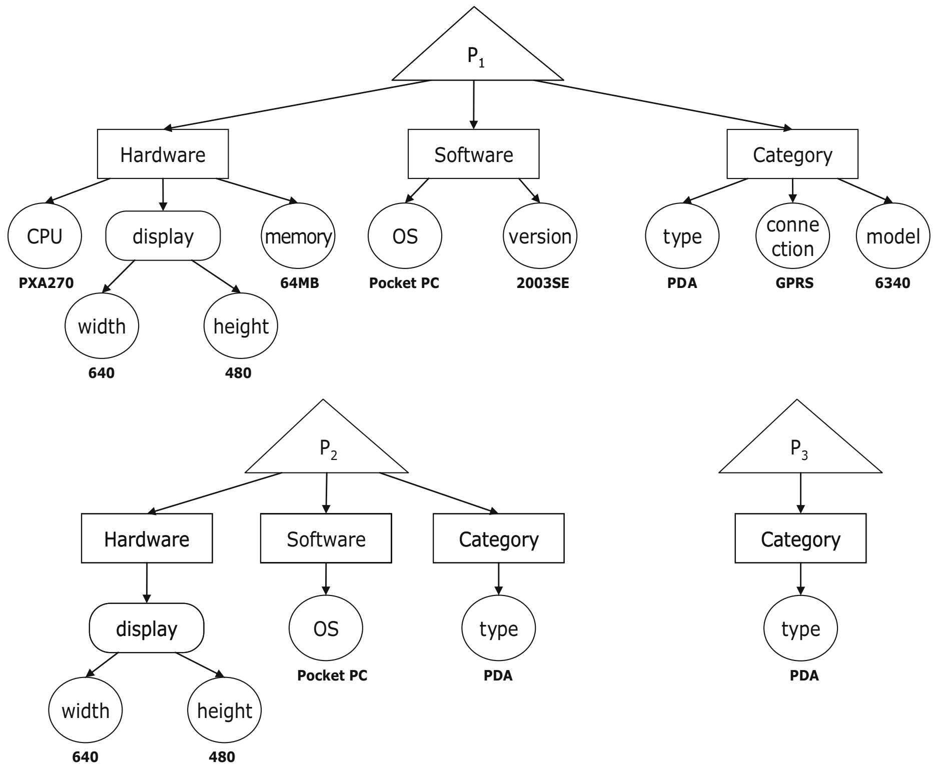


Figure 3 Subsumption between profiles.

in P' such that A_i is covered by A_j. The subsumption relationship is then defined as follows.

Definition 2 (Subsumption) Given two profiles P_1 and P_2, we say that P_1 is subsumed by P_2, in symbols $P_1 \lhd P_2$, if for each dimension D of P_1 there is a dimension D' of P_2 such that for each attribute A of D there is an attribute A' of D' that covers A.

As an example, given the profiles reported in Figure 3, we have that $P_3 \lhd P_2 \lhd P_1$.

Indeed, it turns out that $\lhd$ is a partial order relationship over profiles, as it is reflexive, antisymmetric and transitive.

3.2 Abstract configurations

The notion of configuration is used in our approach to describe, in abstract terms, a suitable adaptation of a Web interface. As we have said in Section 2, in a data-intensive WIS the adaptation process should operate separately on its three main components: content, navigation, and presentation. According to this observation, a configuration has three main components: the *content view* (for adapting the content), the *hypertext definition* (for adapting the navigation), and the *logical style sheet* (for adapting the presentation). They are defined in more detail as follows.

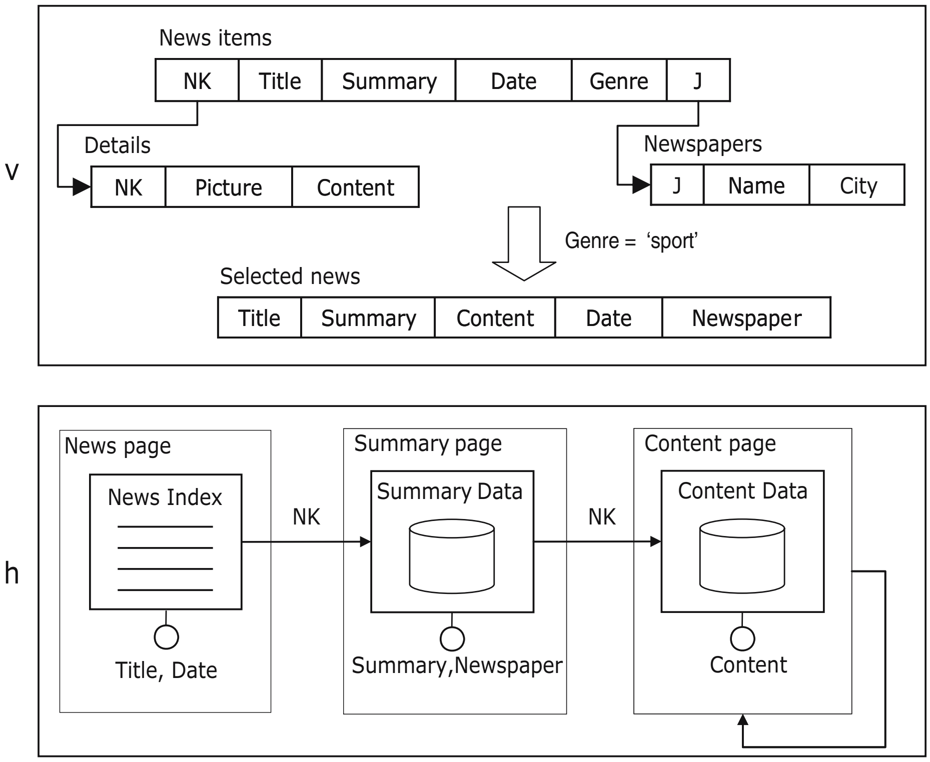
- The *content view* corresponds to a selection of contents extracted (dynamically) from the underlying database (repository) and is simply described by a list of atomic elements and a conditional formula over these elements. More precisely, let d be a database of contents composed by a set of elements (e.g. relational tables) each of which has an associated set of attributes. A content view v over d has the form $\{T, S, F\}$ where:
  - T is the *target* and is composed of a set of attributes of the elements occurring in S;
  - S is the *source* and is composed of a set of elements occurring in d;
  - F is the *conditional formula* and is composed by a conjunction of atoms of the form $A\theta B$ or $A\theta c$, where A and B are attributes of the elements occurring in S, c is a constant value, and θ is a comparison operator.

- The *hypertext definition* describes a way to organize the elements of the content view into a hypertext and is based on a set of constructs taken from WebML [9]. Basically, a hypertext definition is a set of *pages*, which is composed in turn of a set of *units*. A unit represents an atomic portion of a page and includes a direct reference to the elements of a content view from which it takes data. Units can be of different *types* (e.g., an index or just plain content) and can be connected by *links*, in order to obtain a hypertext structure. Links bring parameters from one unit to another, a necessary requirement for a correct navigation through the hypertext. More precisely, a hypertext definition h is a couple $\{P; L\}$ where:
  - P is a set of pages $\{p_1, \ldots, p_n\}$. Each page p_i has a unique name and contains a list of units $[u_1, \ldots, u_n]$. A unit has also a unique name and is associated with: (1) a *type*, belonging to a predefined set of types, (2) a collection of *attributes* occurring in T, and (3) an *ordering* defined over these attributes.
  - L is a set of links $\{l_1, \ldots, l_n\}$. A link l_i has a unique name and is associated with: (1) a couple (u, u') of units, and (2) an attribute occurring in T called *parameter*.

- The *logical style sheet* (or simply *lls*) describes the layout of the pages mentioned in a hypertext definition. It is based on 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. Each attribute is associated with a domain of possible values for the attribute. Given a set $W = \{w_1, \ldots, w_n\}$ of wots, a logical style sheet for W is a function that associates, with each attribute of every wot in W, a value taken for its domain.

Definition 3 (Configuration) An *(abstract) configuration* is a triple $C = (v, h, s)$ where v is a content view, h is a hypertext definition over v, and s is a logical style sheet over h.

All the components of a configuration are abstract notions that can be represented both in a graphical and in a textual way. We do not present the complete syntax of



s

Text	Link	Image
Font: Arial	**Note:** False	**Format:** jpeg
Size: 10pt	**Font:** Arial	**Size:** 178 X 208
Style: Normal	**Size:** 10pt	**Border:** 0pt
Color: Black	**Style:** Underline	**Color:** True
Border: 0pt	**Color:** Blue	**Alignment:** left

Figure 4 A configuration $C = (v, h, s)$.

these representations, since they are very intuitive. An example of a graphical representation of a configuration $C = (v, h, s)$, representing the adaptation illustrated in Figure 1, is given in Figure 4. In this figure: (1) v has been represented by the source, the target and by an arrow labelled with the selection condition, (2) h by a site view (according to the representation of hypertexts proposed by [9]), and (3) s by a table.

A text-based representation of the components of C is reported below.

$$\begin{aligned} v = \{\ & NK, Ti, Su, Co, Da, Ne \mid \\ & \text{NewsItems}(NK, Ti, Su, Da, Ge, J), \text{Details}(NK, Pi, Co), \\ & \text{Newspapers}(J, Na, Ci) \mid Ge = \text{Sport}\ \} \end{aligned}$$

h = { **Page** News [**IndexUnit** NewsIndex(**Attributes** Title, Date; **OrderBy** Date)],
  Page Summary [**DataUnit** SummaryData(**Attributes** Summary, Newspaper)],
  Page Content [**DataUnit** ContentData(**Attributes** Content)];
  Link NewsToSummary(**From** NewsIndex **To** SummaryData; **Parameter** NK),
  Link SummaryToContent(**From** SummaryData **To** ContentData;
    Parameter NK),
  Link ContentToRestOfContent(**From** ContentData **To** ContentData;
    Parameter NK) }

s = { **Text**(**Font**: Arial,..., **Border**: 0pt), **Link**(**Note**: False,..., **Color**: Blue),
  Image(**Format**: jpeg,..., **Alignment**: left) }

It is important to note that a configuration is indeed an abstract notion that can be represented and implemented in several ways. 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 or JSP statements at the navigation level, and a set of CSS files at the presentation level.

As a final comment, we point out that we make use, intentionally, of a simple model for the representation of contexts and Web configurations. This will allow us to keep the notation straight and simplify the presentation of the adaptation methodology, avoiding nonessential details. This fact however does not limit the generality of the approach. In fact, we have shown that our context model can be mapped to other similar formalisms [12]. At the same time, it is possible to show that the expressive power of our notion of configuration is comparable with the core of other Web models, such as WebML [9].

4 Automatic adaptation

In this section we present a rule-base methodology for the automatic generation of a suitable configuration for a given context.

4.1 Adaptation rules

Intuitively, a configuration *matches* a profile if it meets the adaptation requirements of the profile. In our approach, we express the matching relationship between configurations and profiles by means of a novel notion of adaptation rule.

Definition 4 (Adaptation rule) An *adaptation rule* has the form $P_r : C_d \Rightarrow C_f$ where

- $P_r : C_d$ is the *head* of the rule and is composed by:
  - *A parametric profile* P_r, that is, a profile in which variables can appear in place of values,
  - A condition C_d made out of a conjunction of atoms of the form $X \theta Y$ or $X \theta c$, where X and Y are parameters occurring in P_r, c is a constant value, and θ is a comparison operator,

- C_f is the *body* of the rule and is composed by a *parametric configuration*, that is, a configuration in which parameters occurring in P_r can appear in place of values.

The intuitive semantics of an adaptation rule is the following: if the client has a profile P_r and the condition C_d is verified to hold then generate the configuration C_f to obtain the suitable adaptation.

An example of an adaptation rule for a hardware profile P_h is the following:

$R_1 = P_h$(ImageCapable : false, ScreenSize : Y, ColorCapable : Z) : $Y < 1500 \Rightarrow$
({ NK, Ti, Su, Da, Co | NewsItems(NK, Ti, Su, Da, Ge, Ju), Details(NK, Pi, Co)
| true},
{ **Page** News [**IndexUnit** NewsIndex(**Attributes** Title, Date; **OrderBy** Date)],
Page Details [**DataUnit** ContentData(**Attributes** Title, Date, Content)];
Link NewsToDetails (**From** NewsIndex **To** ContentData **Parameter** NK) },
{ **Text**(**Color**: Black, **Size**: 8pt), **Link**(**Style**: Underline, **Color**: Black),
Image(**Size**: $Y \times 0.5$, **Color**: Z) })

Note the use of the parameters Y and Z to resize the images and to eliminate/maintain colors.

A profile can *activate* an adaptation rule and this causes the generation of a configuration. This is made precise by the following definition, in which we call substitution a function that maps variables to constants.

Definition 5 (Rule activation) A profile P *activates* a rule $R = P_r : C_d \rightarrow C_f$ if there exists a substitution σ of parameters of P_r to constant values such that: (1) P subsumes $\sigma(P_r)$ (in symbols $\sigma(P_r) \lhd P$), and (2) $\sigma(C_d)$ is true. In this case, we say that R *generates* the configuration $\sigma(C_f)$ from P.

Consider, for instance, the rule R_1 above and the following profile for a device d:

P_d(Model : 8400, ImageCapable : false, ScreenSize : 1000, OS : GPE,
ColorCapable : false)

Now let σ be the substitution that maps Y to 1000 and Z to false: we have that P_d subsumes $\sigma(P_h)$ and the substitution makes true the condition of the rule. It follows that P_d activates R_1 and this rule generates the following configuration:

v = {NK, Ti, Su, Da, Co | NewsItems(NK, Ti, Su, Da, Ge, Je), Details(NK, Pi, Co)
| true}
h = { **Page** News [**IndexUnit** NewsIndex(**Attributes** Title, Date; **OrderBy** Date)],
Page Details [**DataUnit** ContentData(**Attributes** Title, Date, Content)];
Link NewsToDetails(**From** NewsIndex **To** ContentData; **Parameter** NK) }
s = { **Text**(**Color**: Black, **Size**: 8pt), **Link**(**Style**: Underline, **Color**: Black),
Image(**Size**: 500, **Color**: false) }

Note that a profile P can activate a rule (and thus generate a configuration) defined for a profile that is more general than P. In the example at hand, a profile P_d

for a specific model (8400) activates a rule that does not refer to any model, but it is based on a parametric profile that is "compatible" with P_d. It follows that the same rule can be used for different profiles (possibly not known in advance) that share the same general properties. For instance, we can define a general rule for PDAs that can be used for all devices of this type. At the same time, we can also define a more specific rule that is activated only by a given model of PDA.

Clearly, if a profile P activates two rules R and R' and the profile of R is more "similar" to P than that of R', then only R should be actually activated by P. This is achieved by assuming that, in this case, R *deactivates* R'.

Definition 6 (Rule deactivation) If a profile P activates two rules $R = P_r : C_d \rightarrow C_f$ and $R' = P'_r : C'_d \rightarrow C'_f$ and it is the case that $\sigma'(P'_r) \lhd \sigma(P_r) \lhd P$, then R *deactivates* R'. In this case, R' does not generate any configuration from P.

Consider again rule R_1 above and assume to have another adaptation rule R' having as head: $P'_h(\text{ScreenSize} : X, \text{ColorCapable} : \mathsf{false}) : X > 500$. Then, this rule is activated by the profile P_d reported above but is deactivated by rule R_1, since, after the substitution, the profile of R_1 subsumes that of R'. On the other hand, it is easy to see that a rule R'' with head $P''_h(\text{ScreenSize} : X, \text{OS} : \mathsf{GPE}) : X < 2000$ is activated by P_d but it is not deactivated by R_1, since, after the substitutions, their profiles are not comparable.

It is possible to show that the activation-deactivation process is always deterministic. This result easily follows from the fact that $\lhd$ is a poset.

4.2 Composition of configurations

In our approach, a context is a set of profiles, each of which describes the requirements of an autonomous aspect of the context. On the basis of the activation mechanism described in the previous section, it follows that several configurations can be generated for a given context.

In order to combine the various configurations generated by rule evaluation, we introduce next a special composition operation $\oplus$ over configurations. An important aspect to point out is that this operator defines a *prioritized* composition over two configurations C_1 and C_2. More specifically, if we combine them with the operation $C_1 \oplus C_2$ and some components of C_1 conflict with the corresponding components of C_2 (e.g., for a different organization of the hypertext structure), then, intuitively, the choices done in C_1 are preferred to the choices of C_2. This property will be used by applying the $\oplus$ operation according to a priority order defined over the adaptation rules.

All of this is made more precise by the following definition.

Definition 7 (Composition operator) Given a pair of configurations $C_1 = (v_1, h_1, s_1)$ and $C_2 = (v_2, h_2, s_2)$, $C_1 \oplus C_2$ is a configuration $C = (v, h, s)$ defined as follows:

- Let $v_1 = \{T_1, S_1, F_1\}$ and $v_2 = \{T_2, S_2, F_2\}$; then $v = \{T, S, C\}$ is a content view defined as follows:
  - $T = T_1$ if $T_1 \cap T_2 = \emptyset$ and $T = T_1 \cap T_2$ otherwise;
  - $S = S_1$ if $S_1 \cap S_2 = \emptyset$ and $S = S_1 \cap S_2$ otherwise; and

  - F contains: (1) all the atoms in F_1, and (2) all the atoms in F_2 involving attributes occurring in S that do not occur in atoms in F_1;

- Let $h_1 = \{P_1, L_1\}$ and $h_2 = \{P_2, L_2\}$; then $h = \{P, L\}$ is an hypertext definition defined as follows:
  - P contains all the pages in P_1 and all the pages in P_2 that do not occur in P_1, and
  - L contains all the links in L_1 and all the links in L_2 between pages in P that do not occur in L_1;
- s is a logical style sheet defined as follows:
  - $s(w_i) = s_1(w_i)$ if w_i is a wot occurring in s_1, and
  - $s(w_i) = s_2(w_i)$ otherwise (that is, if w_i is a wot occurring only in s_2).

As an example, consider the adaptation rule R_1 reported in Section 4.1 and the following rule for a user profile u:

$$\begin{aligned}
&R_2 = P_u(\text{PreferredGenre} : X, \text{PreferredFont} : Y) : \mathsf{true} \Rightarrow \\
&(\ \{NK, Ti, Su, Da, Co, Ne, Pi \mid \text{NewsItems}(NK, Ti, Su, Da, Ge, J), \\
&\quad \text{Details}(NK, Pi, Co), \text{Newspapers}(J, Na, Ci) \mid Ge = X\}, \emptyset, \\
&\quad \{\textbf{Text}(\textbf{Font} : Y, \textbf{Size} : \mathsf{12pt})\ \}\ )
\end{aligned}$$

Assume also that rule R_1 (for the device) takes precedence over rule R_2 (for the user). Then, the profiles of the context:

$$\begin{aligned}
C = \{\ &P_d(\text{Model} : \mathsf{8400}, \text{ImageCapable} : \mathsf{false}, \text{ScreenSize} : \mathsf{1000}, \\
&\qquad\qquad \text{ColorCapable} : \mathsf{false}), \\
&P_u(\text{PreferredGenre} : \mathsf{Sport}, \text{PreferredFont} : \mathsf{Arial})\ \}
\end{aligned}$$

activates these two rules and the combination of the configurations generated from them, based on the $\oplus$ operator, produces the final configuration $C = (v, h, s)$ where:

$$\begin{aligned}
v = \{&NK, Ti, Su, Da, Co \mid \text{NewsItems}(NK, Ti, Su, Da, Ge, J), \\
&\quad \text{Details}(NK, Pi, Co) \mid Ge = \mathsf{Sport}\ \} \\
h = \{&\textbf{Page}\ \text{News}\ [\textbf{IndexUnit}\ \text{NewsIndex}(\textbf{Attributes}\ \text{Title, Date};\ \textbf{OrderBy}\ \text{Date})], \\
&\textbf{Page}\ \text{Details}\ [\ \textbf{DataUnit}\ \text{ContentData}(\textbf{Attributes}\ \text{Title, Date, Content})\ ]; \\
&\textbf{Link}\ \text{NewsToDetails}(\textbf{From}\ \text{NewsIndex}\ \textbf{To}\ \text{ContentData};\ \textbf{Parameter}\ \text{NK})\} \\
s = \{\ &\textbf{Text}(\ \textbf{Font}: \mathsf{Arial}, \textbf{Color}: \mathsf{Black}, \textbf{Size}: \mathsf{8pt}), \\
&\textbf{Link}(\textbf{Style}: \mathsf{Underline}, \textbf{Color}: \mathsf{Black}), \textbf{Image}(\textbf{Size}: \mathsf{500}, \textbf{Color}: \mathsf{false})\ \ \}
\end{aligned}$$

Note for instance that just the sport items, which are of interest for the user, are returned but only some attributes of them, as specified in the rule for the device. Moreover, the final font is that preferred by the user, but the size has been set the rule for the device.

4.3 The basic adaptation methodology

We now set the various notions and methods presented in the previous sections into a general methodology for content delivery adaptation.

First of all, we need to fix:

- A basic configuration C, which describes the default Web interface of our information system,
- An initial set $\mathbf{R}$ of adaptation rules, each of which captures the criteria of adaptation for a basic profile, and
- A precedence partial order on the rules in $\mathbf{R}$.

The configuration C should be designed for a context with no specific requirements or constraints.

The overall process of adaptation based on the notions of profile, configuration and adaptation rule can be summarized as follows.

1. The context of the client is captured and represented in terms of a set of profiles, one for each coordinate of adaptation. All the profiles are expressed in terms of the model presented in Section 3.1. As usual, some component 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 from the analysis of his/her navigation).
2. For each profile P of the context, all the configurations generated by rules in $\mathbf{R}$ activated by P (if any) are collected into a set $\mathbf{C}$.
3. The basic configuration C is added to $\mathbf{C}$ (this guarantees that $\mathbf{C}$ contains at least one configuration).
4. All the configurations in $\mathbf{C}$ are merged into a unique configuration C' by means of the composition operator $\oplus$, taking into account the preference order between rules.
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 used at the various levels.
6. The adaptation statements are executed by the underlying systems and the final response is generated.

An relevant aspect of this methodology is that it is easily extensible. First, we assume that the scheme of the context in input (step 1) is not fixed, but additional dimensions and attributes of known profiles or further profiles for new coordinates (e.g., time, location and so on) can be added to the scheme. We also assume that these profiles can be expressed in different languages (such as HTTP header, XML files over specific DTD's, RDF, CC/PP). To this end, we have recently proposed a technique for the management and the translation of heterogeneous context information, expressed in different formalisms [12]. Second, the mechanism for the generation of the adaptation (steps 2–5) allows the designer to improve the adaptation capabilities by simply adding new rules that take into account new components of the context or are more specific for a profile or a class thereof. Actually, the set $\mathbf{R}$ can contain initially a very small set of general rules and can be later enriched as soon as new configurations for new profiles are designed. Finally, the use of logical models for the representation of: (1) the context, (2) the adaptation to be undertaken, and (3) the effect of the adaptation, guarantees that different implementation choices can be made. In particular, the final adaptation statements (step 5) 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.

5 A practical implementation

In this section we illustrate the design of a system implementing our general adaptation methodology.

5.1 An architecture of reference

A very general architecture of a system able to implement the proposed methodology of adaptation is reported in Figure 5. 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,
- A Response Generator (RG), able to generate all the components of a response to deliver over the Web according to a configuration 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 context) and generate a suitable configuration to be sent to the Response Generator.

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, and (2) generate a suitable configuration on the basis of the methodology proposed in Section 4.3. As we have said before, 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 contexts and according to orthogonal requirements of adaptation, possibly not fixed in advance.

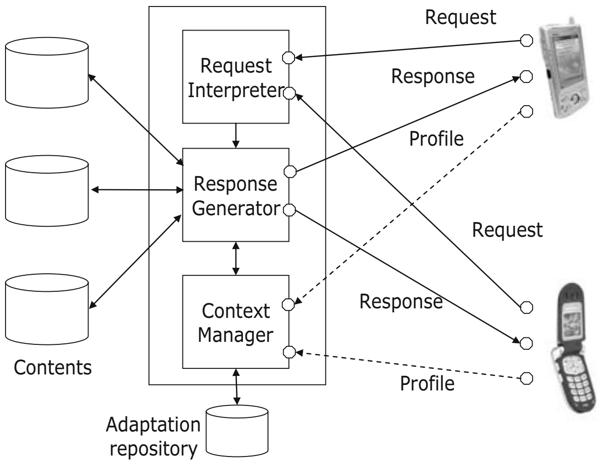


Figure 5 A general architecture of reference.

Figure 6 An extensible context manager.

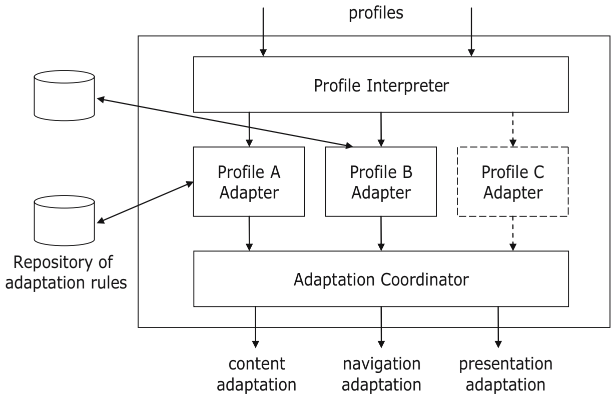
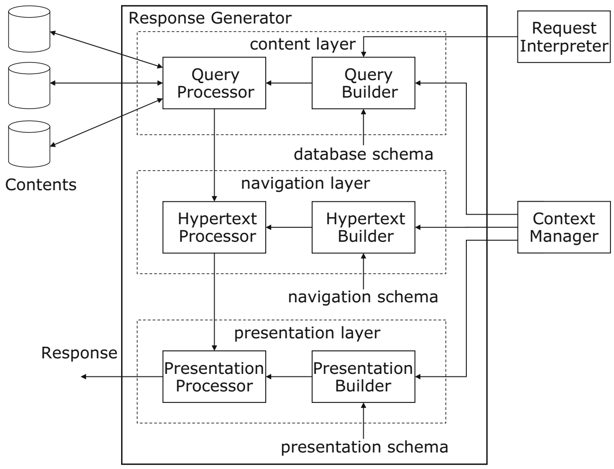


Figure 6 reports a possible architecture for the Context Manager that can meet these requirements. The basic component of this module is the Profile Interpreter, which should be able to get and identify possibly heterogeneous context representations (e.g., CC/PP, XML, HTTP headers) and translate them into a uniform representation expressed in the model proposed in Section 3. As we have said before, we have addressed this problem recently and we have proposed a general technique for the management of heterogeneous context information [12]. According to the methodology presented in the previous section, the various profiles included in the context (e.g., the device characteristics, the user preferences, the location, etc.) are taken as input by a series of modules, one for each profile of the context. The main task of these modules is to generate a uniform set of adaptation specifications, expressed in terms of a configuration that satisfies the specific requirements of a profile. According to our approach, this work is supported by a repository of adaptation rules. Actually, the various adapters are similar and work in the same way.

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 takes into account the various adaptation requirements and can

Figure 7 A response generator.



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 7), one for each component of a configuration. The first module combines the content view 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 navigation scheme of the Web site (e.g., by splitting pages or adding links) to satisfy the requirements of adaptation specified by the hypertext definition of the selected configuration. Finally, the last module is in charge of implementing the logical style sheet of the configuration at hand.

5.2 An optimization strategy

In a real world scenario, the number of clients with different requirements of adaptation can increase rapidly. In order to guarantee an effective responsiveness of our system to these adaptation requirements, the number of rules should likewise increase quickly. Since the rule evaluation technique requires the matching of the profiles of each client with all the available rules, it follows that performance and scalability of the system becomes soon a challenge.

For this reason, we have defined an optimization strategy based on a clustering technique over the adaptation rules: each cluster represents a class of rules that match the requirements of similar contexts.

The technique is based on a metric distance between profiles that takes advantage of the tree structure of the generic profile. We get inspiration from the *tree edit distance* [32], where the distance between two trees T_1 and T_2, denoted by $\delta(T_1, T_2)$, depends on the sequence of operations (taken from a predefined set) that allows us to transform T_1 into T_2. By assigning a fixed cost to each operation, the distance simply corresponds to the sum of costs of the operations in the sequence. In the tree representation of our notion of profile, we distinguish two types of nodes: dimension and attribute. We have therefore operations to modify, remove or create dimensions and attributes nodes. The cost of an operation o is denoted by $\gamma(o)$. Given two trees T_1 and T_2, an *s-derivation* S from T_1 to T_2 is a sequence of operations $o_1, \ldots, o_k$ that transforms T_1 into T_2. We define the cost $\gamma(S)$ of S as $\sum_{i=0}^{k} \gamma(o_i)$.

Definition 8 (Distance between profiles) Given two profiles P_1 and P_2, the distance between P_1 and P_2, denoted by $\delta(P_1, P_2)$, corresponds to $\gamma(S)$, where S is the s-derivation from P_1 to P_2 with minimum cost.

In our approach, a cluster contains the profiles occurring in the adaptation rules and is organized as a tree. Each of these profiles contains a link to the corresponding rule. In this tree a profile P_1 is parent of a profile P_2 if $P_1 \lhd P_2$, therefore the root of a cluster represents the most general profile in the cluster.

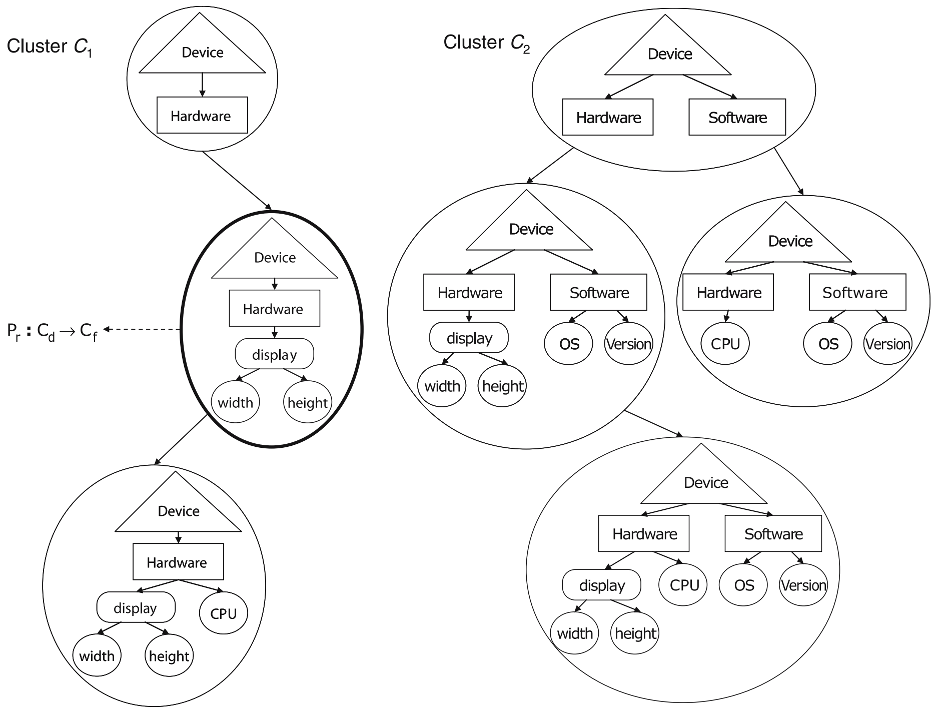


Figure 8 Clusters of profiles.

The choice of the roots of the clusters is left to the designer. Then, given an initial set of clusters, a profile P of an adaptation rule $P : C_d \rightarrow C_f$ is inserted into a cluster as follows.

- We select the cluster $\mathcal{C}$ having as root a profile P' such that (1) $P' \lhd P$, and (2) the distance between P and P' is minimum; if there is no cluster whose root subsumes P then we introduce a new cluster having P as root;
- Starting from the root of $\mathcal{C}$, we insert P in $\mathcal{C}$ as follows:
  - If there is no child of the root P' of $\mathcal{C}$ such that $P' \lhd P$, then (1) P becomes the child of P', and (2) each child P''' of P' such that $P \lhd P'''$ becomes child of P';
  - Otherwise, we insert P in the sub-tree $\mathcal{C}'$ of $\mathcal{C}$ having as root the child P'' of P' such that (1) $P'' \lhd P$, and (2) the distance between P and P'' is minimum;

Once the set of rules has been clustered in this way, given a new profile P describing a new client of our system, the search of the rules activated by P can be efficiently performed by applying a search technique similar to the insertion mechanism described above: when we find a profile P' in a cluster such that no child of P' subsumes P, then we activate the rule linked to P'. If the search fails, the basic configuration is returned.

As an example, let us assume to have the clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ reported in Figure 8. Let us denote with α, β and χ the operations to insert, delete and modify a node of a profile in its tree representation (i.e., a dimension, a complex attribute or a simple

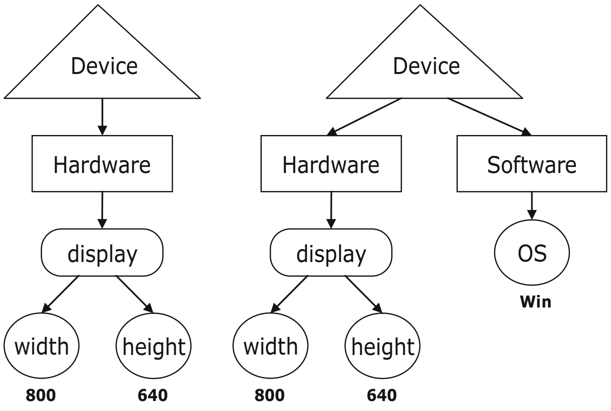


Figure 9 Profiles of devices.

attribute of the profile). Moreover, let us assume that the cost function is defined as follows.

γ	α : insert	β : delete	χ : modify
Dimension	3	4	2
Complex attribute	2	3	1
Simple attribute	1	1	1

Let us now consider the profile P_1 reported in left hand side of Figure 9: according to the search method described above, to identify the rules activated P_1, we first select the cluster having as root a profile P' such that $P' \lhd P_1$ and the distance from P_1 is minimal. Now P_1 subsumes both the roots of $\mathcal{C}_1$ and $\mathcal{C}_2$: the s-derivation needed to transform P_1 into the root of $\mathcal{C}_1$ is $S_1 = \langle\beta(\text{display}), \beta(\text{width}), \beta(\text{height})\rangle$ with a cost $\gamma(S_1) = 5$, whereas the s-derivation needed to transform P_1 into the root of $\mathcal{C}_2$ is $S_2 = \langle\beta(\text{display}), \beta(\text{width}), \beta(\text{height}), \alpha(\text{software})\rangle$ with a cost $\gamma(S_2) = 8$. Then, cluster $\mathcal{C}_1$ is selected and since the child of the root coincides with P_1, the corresponding rule is activated.

A possible rule for this profile can be the following:

$R = P_d(\text{Hardware}(\text{display}(\text{width} : Y, \text{height} : Z))) : Y > 500 \wedge Z > 300 \Rightarrow$
({ NK, Ti, Su, Da, Co | NewsItems(NK, Ti, Su, Da, Ge, Ju), Details(NK, Pi, Co) | true},
{ **Page** News [**IndexUnit** NewsIndex (**Attributes** Title, Date; **OrderBy** Date)],
Page Details [**DataUnit** ContentData(**Attributes** Title, Date, Content)];
Link NewsToDetails (**From** NewsIndex **To** ContentData **Parameter** NK) },
{ **Text**(**Color**: Black, **Size**: 8pt), **Link**(**Style**: Underline, **Color**: Black),
Image(**Size**: $Y \cdot 0.5 \times Z \cdot 0.5$, **Color**: true) })

Let us now consider the profile P_2 reported in the right hand side of Figure 9. In this case, it is easy to see that the cluster whose root subsumes P_2 and has the lowest distance from P_2 is $\mathcal{C}_2$. By visiting this cluster, we have that no node subsumes P_2 and therefore the rule associated with the profile in the root of $\mathcal{C}_2$ is activated.

It turns out that the clustering technique correctly identify the rules to be activated and can reduce greatly the number of rules that need to be accessed. This is confirmed by the experimental results that will be presented in Section 5.4.

5.3 FAWIS

The methods and techniques illustrated in this paper have been implemented in FAWIS (short hand for Flexible Adaptation of Web-based Information Systems), an experimental tool for the adaptation of content delivery in Web Information Systems. FAWIS has been developed at Roma Tre University within MAIS (www.mais-project.it), a research project funded by Basic Research Funds (FIRB Program) of the Italian Department of University and Research (MIUR).

The architecture of FAWIS is based on the general scheme discussed in Section 5.1 and is reported in Figure 10. This architecture also includes a Session Manager, not shown in the figure, that is able to infer the interests of the user on a set of Web pages according the chronology of his/her navigation.

We have equipped the Context Manager with an *Adaptation Designer* that allows the designer to extend the functionality of the Context Manager by: (1) enriching the repository of adaptation rules for an existing adaptation module and (2) building a new adaptation module to satisfy the requirements of adaptation of a new coordinate. In order to better support the execution of these tasks, the Adaption Designer is equipped with a user-friendly interface, which is reported in Figure 11. Note that adaptation rules and configurations can be specified in a graphical way.

An important component of FAWIS is the profile interpreter that is currently able to capture profiles expressed in different languages (including HTTP headers, UAProf, CC/PP, and plain XML) making use of a technique illustrated in [12]. These profiles are translated into a homogeneous internal representation, based on our context model, and are given as input to the various adaptation modules.

Currently, three adapters have been implemented, devoted to the management of device capabilities, user preferences, and network characteristics, respectively. A repository of adaptation rules supports the task of each adapter. In particular, the

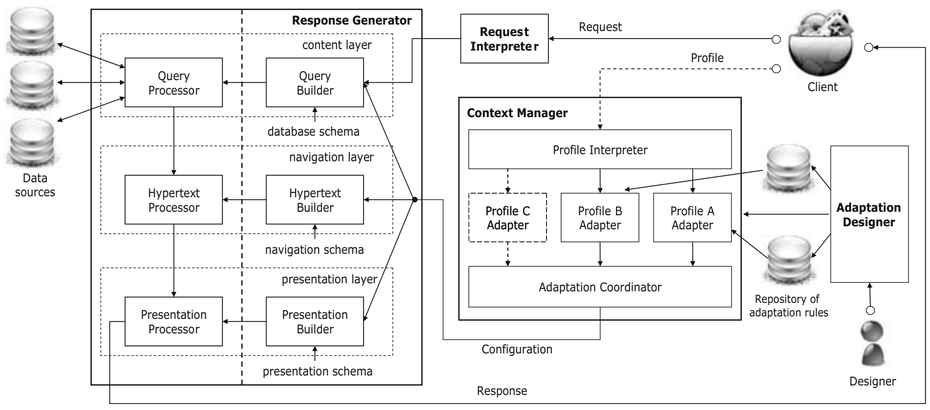


Figure 10 FAWIS architecture.

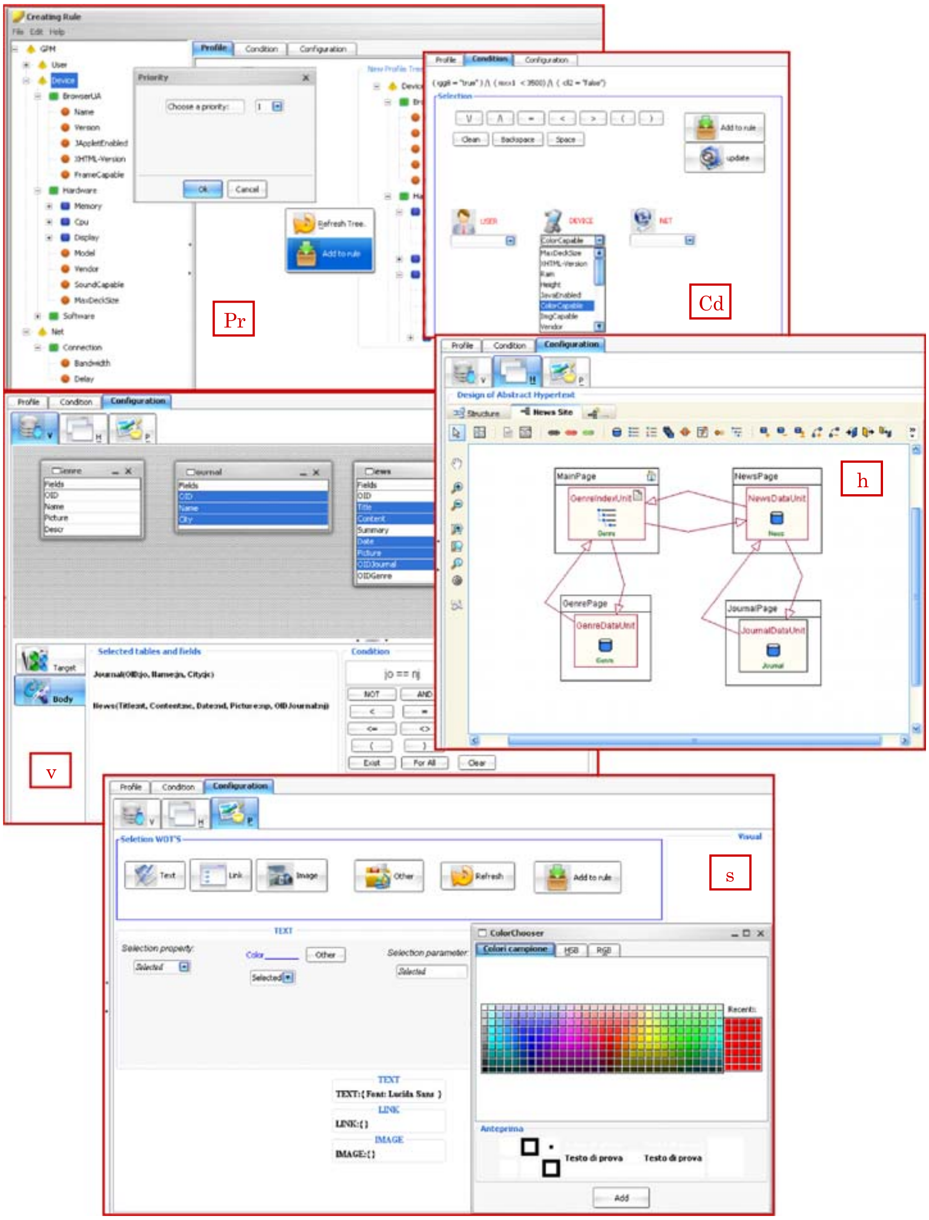


Figure 11 Design of adaptation rules.

repository for the devices has been built by using the WURFL database (available at wurfl.sourceforge.net), which includes information about capabilities and features of a large set of mobile devices currently available on the market. Each adapter takes as input a profile and selects the rules to activate by adopting the method described in Section 4 and the optimization strategy described in the previous section. Then, the

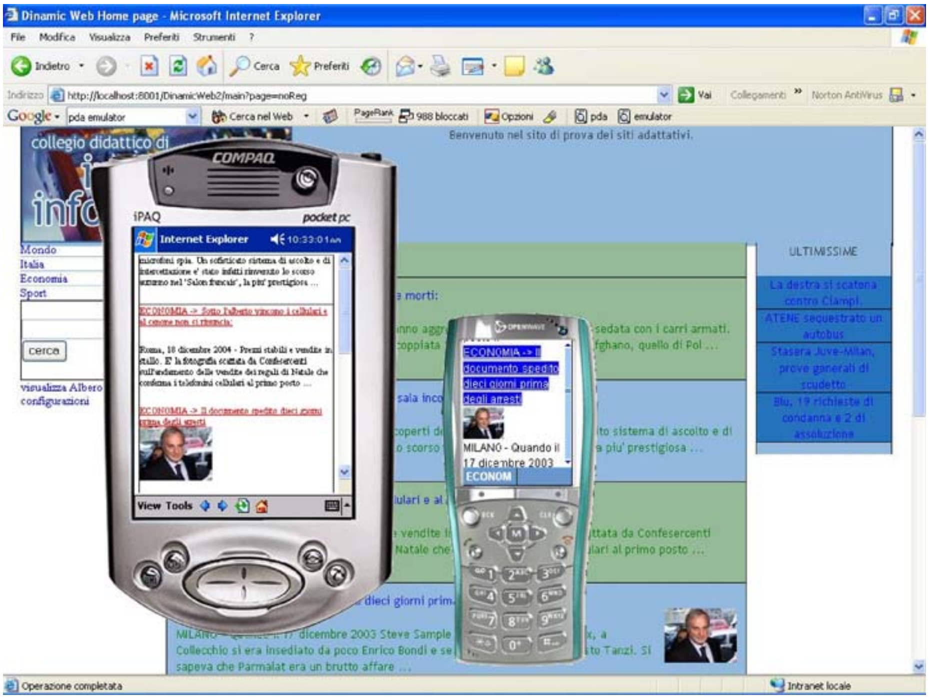

Dinamic Web Home page - Microsoft Internet Explorer
http://localhost:8001/DinamicWeb2/main?page=noReg
Benvenuto nel sito di prova dei siti adattativi.
collegio didattico di
COMPAQ
iPAQ
pocket pc
Internet Explorer
Mondo
Italia
Economia
Sport
cerca
visualizza Albero
configurazioni
ULTIMISSIME
La destra si scatena contro Ciampi.
ATENE sequestrato un autobus
Stasera Juve-Milan, prove generali di scudetto
Blu, 19 richieste di condanna e 2 di assoluzione
ECONOMIA -> Il documento spedito dieci giorni prima degli arresti
MILANO - Quando il 17 dicembre 2003
View Tools
Operazione completata
Intranet locale

Figure 12 FAWIS at work.

Adaptation coordinator merges the configurations generated by the activated rules and returns a configuration that describes the final adaptation.

Figure 12 reports the final results of the adaptation process performed by FAWIS over the same information content, according to three contexts, differing in terms of devices, networks and user preferences.

FAWIS is implemented in Java and makes use of Jena (jena.sourceforge.net), an RDF based framework, equipped with an inference engine, for the development of Web applications [19, 24]. The system relies on a relational database of contents and the selection of data is done by translating content views into SQL queries, hypertext definitions into WML or XHTML statements, and logical style sheets into CCS style sheets. Actually, FAWIS is equipped with its own processors for the generation of the final Web interface, but the ultimate implementation can be demanded to an external system. Indeed, in a definitive architecture, FAWIS should reside on a proxy server to alleviate the server workload.

5.4 Experimental results

We have tested the effectiveness and the efficiency of FAWIS in several practical cases. On the server side, we have used an IBM computer xSeries 225, equipped with a Xeon2 2.8 Ghz processor, a 4 GB RAM, and a 120 GB HDD SCSI. The Web site used in the tests collects a repository of around 100,000 news items, many of which containing multimedia contents.

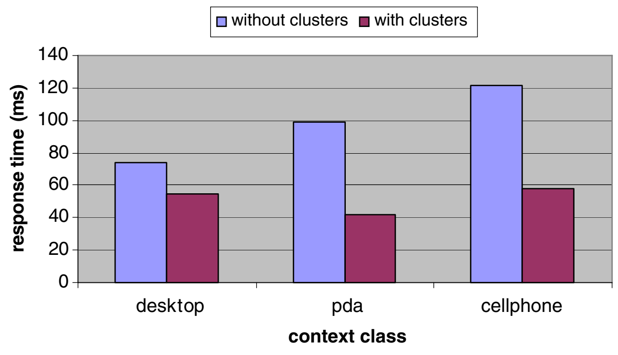


Figure 13 Medium response time in absence and in presence of clusters.

On the client side, the Web site has been accessed by several clients: we have measured the medium response time for several profiles of devices, networks and user preferences.

More specifically:

- The devices have been classified and grouped in three main categories: mid-range desktops, PDAs, and cellular phones. Desktops have been classified according to the processor capabilities (power in MHz), RAM, hard disk, video, and operating system. Cellular phones have been classified according to the supported markup languages (WML, XHTML or HTML), Java support, audio/video/image capabilities and the supported size of MMS. PDAs have been classified according to the processor capabilities, operating system, multimedia capabilities, and supported protocols (WiFi, Bluetooth, GPRS). After the identification of such classes, we have selected representatives of each class by choosing among the devices currently available on the market and produced by the different companies (such as Nokia, Siemens, Motorola, Sony Ericsson, Samsung, HP, Apple).
- We have then generated a large variety of profiles for virtual users with different preferences about the content (selection of interesting news), navigation (the most popular web pages) and presentation features.
- Finally, different networks with varying rates of service have been used. In this case we classified the alternatives according to the bandwidth and delay values.

Further details on the tests we have performed are available in [11].

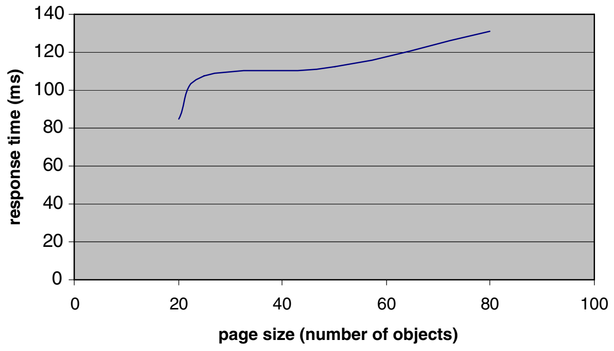


Figure 14 Medium response time related to number of elements in the page.

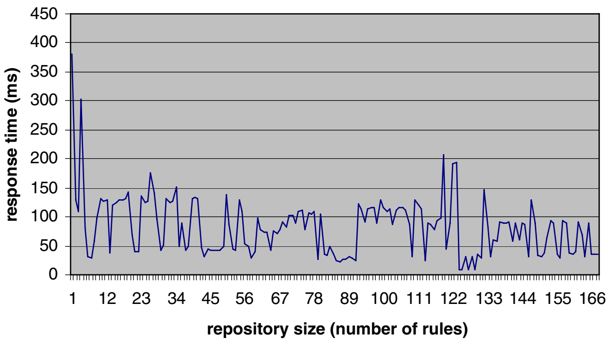


Figure 15 Medium response time related to number of adaptation rules.

In general, the results obtained so far are good. 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" contexts, as described in Section 4.

With respect to the efficiency of the approach, we have performed three main series of tests. In the first series we have measured, for each class, the response time of the FAWIS in absence and in presence of clusters of rules. Then, we have analyzed the response time with respect to the complexity of the page to deliver. Finally, we have tested the performance of the system with respect to the number of adaptation rules stored in the repository.

The results of our experiments are synthesized in three diagrams, as follows.

- Figure 13 reports the average response time (vertical axis) of each main class of profiles (horizontal axis). The diagram shows a good performance in the generation of the adapted response. The desktop profiles present an average time of 72 ms, the PDA profiles 97 ms and the cellular phone profiles 121 ms. The diagram also compares the medium response time in absence and in presence of clusters, in the case of a repository of around 15,000 rules (mainly due to a large set of user profiles). It turns out that, using clusters, the performance of the system improves in general, almost halving the response time in the case of mobile devices (from 121 ms to 59 ms).
- Figure 14 reports the average response time (vertical axis) with respect to the complexity of the pages included in the response (horizontal axis). The complexity has been measured in terms of the number of elements (text, link, image and so on) occurring in a Web page weighted with their sizes. Note that the system is able to provide an answer rather efficiently, stabilizing around 130 ms;
- Figure 15 reports the average response time with respect to the number of rules (in the diagram one unit corresponds to 100). The diagram confirms the effectiveness of the methodology and of the optimization strategy, since the medium response time first decreases and then does not increase with the number of rules (the response time starts from 250 ms and stabilizes around 50–90 ms). Note that a relevant number of rules implies a more detailed adaptation and this makes the system more effective.

6 Related work

Being dependent on or aware of time, location, device, user etc. is a facet of the new-generation ubiquitous Web applications that exploit the new possibilities of Web technology. In this paper we have focused on data-intensive Web applications and the adaptation related to content, navigation, and presentation.

A number of approaches and methodologies for the design and development of data-intensive Web applications (WIS) have been proposed in the last years and many have chosen a model-based approach. Typically we observe a model that describes the content (domain) as a first step in the approach. On the basis of that content or domain model the approaches help the designer to formulate the navigation that is going to be realized over that content. WebML [9] for example uses a hypertext model to describe how the data from the domain model is published in a hypertext format, by distinguishing site views, areas, pages and content units. Other methods use similar constructs such as the navigational context that OOHDM [27, 28] uses for making node and link-based views over OO structural descriptions. Hera [30] uses slice units and slice relationships to compose a navigational structure over the data semantically described in RDF(S). The UWA project concentrates on constructing a special purpose design approach to model Web applications [29]. The approach clearly distinguishes different design aspects: requirements elicitation, hypermedia design, transaction design, and customization design. Several other methodologies we could name here [1, 4, 6, 10, 14, 18, 21, 25, 26, 29] and many of them also distinguish the three characteristic aspects in the design we named before: (1) content (domain, data), (2) navigation, (3) presentation, as we have done in our approach. Actually, after constructing the navigation over the content, many of the approaches allow to configure how this conceptual navigation structure is implemented with specific presentation details: the global transformation from content data to presentation on the (mobile) device is thus split in a step geared more towards the semantical organization into a hypertext, followed by a step that considers the layout and rendering of this navigational structure into concrete pages and links for the particular device.

Some approaches and methodologies in this class address the issue of adaptation, i.e. they offer the designer possibilities to configure the particular *context* in which the information is delivered. 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" [31]. These may include the access device, the network QoS, the user preferences, the location, and so on. Dey indicates different interpretations of context [13], while Jameson illustrates how this user and context modeling can play a role in adaptive systems [20]. Actually, many context models have been proposed in the literature (see for instance [7] for a survey). We aim in our approach to provide a generic facility for context and its exploitation in adaptation. Therefore, we make use of a model that does not pretend to be better than others, but just simple and general enough for our purposes. As we have said before however, this fact does not limit the generality of the approach. In fact, we have shown elsewhere that our context model can be mapped to other similar formalisms [12].

We see approaches to adaptation for specifically making Web pages suitable for the mobile device [3, 17], which often concentrate on the presentation design step, approaches for managing mobility [4, 5, 10], with a stronger emphasis on distribution and content integration, and approaches to add adaptation to the general WIS design approaches we described above, which consider adaptation in the navigation design step as well, e.g. to WebML [8] or to Hera [15, 16]. Several adaptation techniques have been proposed and, similarly to our approach, some of them are rule based. In particular, in [2] the authors propose a rule-based mechanism that allows the modification of the context (modelled as a set of attribute/value pairs) to solve conflicts in the adaptation requirements. In the UWE approach [22], an event-condition-action (ECA) technique is proposed to specify the customization of a Web application. Rules are activated by events, such as the change of the bandwidth, rather than by client profiles, as it happens in our approach. However, a precise definition of rule activation is not given. ActiveWeb is an XML-based approach to reconfigure Web pages by means of ECA rules [23]. These rules however refer only to the user behavior and are rather low level. Finally, Ceri et al. propose a method for the adaptation of Web applications that take into account the user interaction. Again, ECA rules are used to specify adaptivity actions, mainly at the navigational level, to be undertaken when events, consisting in a page request, occur [8]. Differently from these approaches, our adaptation rules can be viewed as *production* rules that are activated by a context (involving several characteristics) rather than an event (usually a user action).

In general, we believe that the majority of the above mentioned approaches provide specific solutions that are suited for a particular class of predefined adaptation requirements: most of them consider device characteristics and user preferences and allow the designer to describe the effect of the characteristics and preferences on the navigation and presentation. This makes it not exactly trivial when a new adaptation functionality needs to be added, since the adaptation design facilities need to be extended, but it is not clear how this could be done in a natural way. In other words, in these approaches it is hard to take into account new aspects of the context and modify the adaptation accordingly. That is why we have addressed in this paper the problem from a generic perspective aimed at facilitating a uniform and generic solution for the adaptation desired in the WIS being designed.

7 Conclusions and future work

In this paper we have presented a novel rule-based approach supporting the automatic adaptation of content delivery in Web Information Systems. This problem arises in common scenarios where the information system is accessed, in different contexts, by a variety of mobile devices and users. To guarantee the generality of the approach we have introduced a generic notion of profile, which has been used to model a variety of contexts in a uniform way, and configuration, which has been be used to describe how to build a suitable adaptation. The adaptation is achieved automatically by means of production rules that specify, in a declarative way, how a configuration can be generated to meet the requirements of adaptation of a given profile. Different contexts and orthogonal requirements of adaptation, possibly not fixed in advance, can be taken into account in this process. The approach has been implemented in a system that has been used to test its flexibility and efficiency.

The results presented in this paper are subject of further conceptual and practical investigation. From a conceptual point of view, we are currently investigating the expressive power of rules and we are developing static techniques for testing general properties of rules. From a practical point of view we are improving the efficiency of the system and we are extending its functionality.

Acknowledgements We would like to thank the anonymous referees for their helpful comments.

References

1. Atzeni, P., Merialdo, P., Mecca, G.: Data-intensive web sites: design and maintenance. World Wide Web J. **4**(1–2), 21–47 (2001)
2. Bettini, C., Maggiorini, D., Riboni, D.: Distributed context monitoring for continuous mobile services. In: Proceedings of Working Conference on Mobile Information Systems (MOBIS'05). Leeds (UK) (2005)
3. Bickmore, T., Girgensohn, A., Sullivan, J.: Web page filtering and reauthoring for mobile users. Comput. J. **42**(6), 534–546 (1999)
4. Bolchini, C., Curino, C., Schreiber, F.A., Tanca, L.: Context integration for mobile data tailoring. In: Proceedings of 7th Int. Conference on Mobile Data Management (MDM'06). Nara, Japan (2006)
5. Cabri, G., Leonardi, L., Mamei, M., Zambonelli, F.: Location-dependent services for mobile users. IEEE Trans. Syst. Man Cybern., Part A, Syst. Humans **33**(6), 667–681 (2003)
6. Cachero, C., Gomez, J., Pastor, O.: Object-oriented conceptual modeling of web application interfaces: the OO-HMethod presentation abstract model. In: Proceedings of 6th Int. Conf. on E-Commerce and Web Technologies (EC-Web'01). Munich, Germany (2001)
7. Cappiello, C., Comuzzi, M., Mussi, E., Pernici, B.: Context management for adaptive information systems. Electrical Notes in Theor. Comp. Sci. **146**(1), 69–84 (2006).
8. Ceri, S., Daniel, F., Demaldé, V., Facca, F.M.: An approach to user-behavior-aware web applications. In: Proceedings of 5th Int. Conf. on Web Engineering (ICWE'05). Springer, Sydney, Australia (2005)
9. Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M.: Designing Data-Intensive Web Applications. Morgan Kaufmann, San Mateo, CA (2003)
10. Curino, C., Quintarelli, E., Tanca, L.: Ontology-based information tailoring. In: Proceedings of 2nd Int. Workshop on Database Interoperability (InterDB'06). Atlanta, USA (2006)
11. De Virgilio, R.: A general methodology for context-aware adaptation in web information systems. Ph.D. thesis, School of Computer Engineering, Roma Tre University, Roma, Italy (2006)
12. De Virgilio, R., Torlone, R.: Modeling heterogeneous context information in adaptive web based applications. In: Proceedings of 6th ACM Int. Conference on Web Engineering (ICWE'06). Palo Alto, California (2006)
13. Dey, A.: Understanding and using context. Personal and Ubiquitous Computing Journal **5**(1), 4–7 (2001)
14. Fiala, Z., Hinz, M., Meissner, K., Wehner, F.: A component-based approach for adaptive dynamic web documents. J. Web Eng. **2**(1–2), 58–73 (2003)
15. Fiala, Z., Frasincar, F., Hinz, M., Houben, G.J., Barna, P., Meißner, K.: Engineering the presentation layer of adaptable web information systems. In: Proceedings of 4th Int. Conf. on web Engineering (ICWE'04). Springer, Munich, Germany (2004)
16. Frasincar, F., Houben, G.J.: Hypermedia presentation adaptation on the semantic web. In: Proceedings of 2nd Int. Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH'02). Springer, Malaga, Spain (2002)
17. Gu, W., Helal, A.S.: An XML based solution to delivering adaptive web content for mobile clients. In: Proceedings of Int. Symposium on Performance Evaluation of Computer and Telecomm. Systems (SPECTS'04). San Jose, California (2004)
18. Isakowitz, T., Stohr, E.A., Balasubramanian, P.: RMM: a methodology for structured hypermedia design. Commun. ACM **38**(8), 34–44 (1995)
19. HP-Labs: Jena. http://jena.sourceforge.net/ (2004)
20. Jameson, A.: Modeling both the context and the user. Personal and Ubiquitous Computing Journal **5**(1), 29–33 (2001)

21. Jin, Y., Xu, S., Decker, S.: OntoWebber: Model-driven ontology-based web site management. In: 1st Semantic Web Working Symposium (SWWS'01), pp. 529–547 Stanford, California, USA (2001)
22. Kappel, G., Retschitzegger, W., Schwinger, W.: Modeling ubiquitous web applications: the WUML approach. International workshop on data semantics. In: Int. Workshop on Data Semantics in Web Information Systems (DASWIS'01). Yokohama, Japan (2001)
23. Kiyomitsu, H., Takeuchi, A., Tanaka, K.: Activeweb: Xml-based active rules for web view derivations and access control. In: Int. Workshop on Information Technology for Virtual Enterprises (ITVE'01). Queensland, Australia (2001)
24. McBride, B.: Jena: implementing the RDF model and syntax specification. In: Proceedings of 2nd Int. Workshop on the Semantic Web (SemWeb'01). Hong Kong, China (2001)
25. Pastor, O., Fons, J., Pelechano, V.: A method to develop Web applications from Web-oriented conceptual models. In: Proceedings of 3rd Int. Workshop on Web-oriented Software Technology (IWWOST'03). Oviedo, Spain (2003)
26. Pelechano, V., Fons, J., Albert, M., Pastor, O.: Developing web applications from conceptual models. In: In Proceedings of 15th Conference on Advanced Information Systems Engineering (CAISE'03). Klagenfurt, Austria (2003)
27. Schwabe, D., de Almeida Pontes, R., Moura, I.: OOHDM-Web: an environment for implementation of hypermedia applications in the WWW. ACM SIGWEB Newsletter **8**(2), 18–34 (1999)
28. Schwabe, D., Rossi, G., Barbarosa, S.D.J.: Systematic hypermedia application design with OOHDM. In: Proceedings of 7th ACM Conference on Hypertext (HYPERTEXT'96). Washington, USA (1996)
29. UWA Consortium: The UWA approach to modeling ubiquitous web applications. In: Proc. of IST Mobile & Wireless Telecommunications Summit, Thessaloniki, Greece (2002)
30. Vdovjak, R., Fransincar, F., Houben, G.J., Barna, P.: Engineering semantic web information systems in Hera. J. Web Eng. **2**(1–2), 3–26 (2003)
31. W3C: Device independence principles. http://www.w3.org/TR/di-princ/ (2003)
32. Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput. **18**(6), 1245–1262 (1989)