A Rule-based Approach to Content Delivery Adaptation in Web Information Systems

Roberto De Virgilio, Riccardo Torlone
Università Roma Tre, Rome, Italy
{devirgilio,torlone}@dia.uniroma3.it

Geert-Jan Houben
Vrije Universiteit Brussel, Brussels, Belgium
Geert-Jan.Houben@vub.ac.be

Abstract

In this paper, we present a rule-based approach supporting the automatic adaptation in Web Information Systems. The approach relies on the general notions of profile and configuration. Special rules are used to specify, in a declarative way, how to build a configuration that satisfies the requirements of adaptation for a profile. The rule evaluation technique guarantees that different contexts and orthogonal requirements of adaptation, possibly not fixed in advance, can be taken into account in the adaptation process.

1 Introduction

The number and the spread of mobile devices able to provide access to the Web *everywhere* and *anytime* is increasing day by day. These devices are so different that the problem of delivering Web information in a suitable way involve not only presentational aspects, 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 pages whose dimensions depend closely on specific features of the device. Indeed, this functionality can be generalized by taking into account other aspects characterizing the *context* in which Web information is to be delivered [11]. These may include the user preferences, the network QoS, the location, an so on. It follows that a modern context-aware adaptive system should guarantee a high level of flexibility in terms of: (i) responsiveness to highly changing requirements of adaptation, and (ii) suitable actions to be undertaken to meet these requirements.

Several methodologies have been proposed for the design of Web based Information Systems [3, 9, 8] and many of them address the issue of adaptation by presenting suitable models and techniques [2, 6, 7, 10]. However, the majority of them are actually specific solutions, suited only for predefined adaptation requirements (usually device characteristics and user preferences) and are hardly reusable when a new adaptation functionality needs to be added. In particular, in these approaches it is hard to take into account new aspects of the context and modify the adaptation accordingly.

With this in mind, we have recently proposed a very general methodology for content delivery adaptation in Web based information systems that can be used for different and independent requirements of adaptation, possibly not fixed in advance [4]. The approach relies on two basic notions: the profile and the configuration. The former is used to model a variety of contexts in a uniform way. The latter describes, in abstract terms, how to build the various levels of a Web interface (content, navigation and presentation). In this methodology the specification of a configuration suitable for a profile is left to the designer.

In this paper we extend this methodology and present a general approach capable of generating a suitable adaptation in Web Information Systems in a completely automatic way. To this end, we introduce special rules that allows us to specify, in a declarative way, how to build a configuration to satisfy the requirements of adaptation for a given profile. The rule evaluation technique guarantees that different contexts and orthogonal requirements of adaptation, possibly not fixed in advance, can be taken into account in the adaptation process. This is guaranteed by a rule activation mechanism that takes into account partial matchings and by a method to combine configurations generated by different rules and solve possible conflicts between them.

The rest of the paper is organized as follows. In Section 2, we illustrate the basic notions of profile and configuration and the adaptation methodology. In Section 3, we present the rule-based technique for the automatic generation of configurations and, in Section 4, we draw some conclusions and sketch future work.

2 Profiles, configurations and adaptations

In this section we present an adaptation methodology based on two basic notions: the generic profile and the abstract configuration.

2.1 Generic profiles

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 presentation of its contents). Examples of profiles are the user, the device, the location, and so on. A *dimension* is a property that characterizes a profile. Each dimension can be conveniently 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 has a value associated with it, whereas a complex attribute has a set of (simple or complex) attributes associated with it. For example, the display attribute can be composed by the simple attributes width and height. In this model, a *context* is a collection of profiles.

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 our model, different profiles over the same dimensions 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 the attributes of P_2 at the same or at coarser level of detail.

More precisely, we first say that an attribute A *covers* another attribute B if either they are simple and $A = B$ or they are composite and for each sub-attribute A_i of A there is a sub-attribute B_j of B such that A_i covers B_j. The subsumption relationship is then defined as follows.

Given two profiles P_1 and P_2, we say that P_1 is subsumed by P_2, $P_1 \lhd P_2$, 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' covered by A.

2.2 Abstract configurations

In a data-intensive Web the adaptation process should operate separately on its three main components: content, navigation, and presentation. According to this observation, we introduce the notion of *(abstract) configuration* as a triple $C = (q, h, s)$ where: (i) q is a query over the underlying database expressed in relational calculus, a declarative and abstract language for relational databases [1]; (ii) h is an abstract hypertext definition expressed in WebML [3], a conceptual model for Web application which allows us to describe the organization of Web pages in a tight and concise way, by abstracting their logical features; (iii) s is presentation specification expressed in terms of an original notion of *logical style sheet*, which is a set of tuples over a predefined collection of *Web Object Types (WOTs)* like 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.

As an example, a textual representation of a configuration $C = (q, h, s)$ is the following.

$$
\begin{aligned}
q = \{\, & T : x_1, S : x_2, D : x_3, C : x_4, N : x_5 \mid \\
& \mathit{NewsItems}(N : x_0, T : x_1, S : x_2, D : x_3, G : x_6, J : x_7), \\
& \mathit{Details}(NK : x_0, P : x_8, C : x_4), \\
& \mathit{Newspapers}(J : x_7, N : x_5, C : x_9), x_6 = \mathit{Sport}\}
\end{aligned}
$$

```
h = IndexUnit NewsIndex
      ( source News Items; attributes Title, Date;
       orderby Date )

          DataUnit ContentData
            ( selector Item = CurrentItem;
             attributes Title, Date, Content)
          link NewsToDetails
            ( from NewsIndex to ContentData
             parameters CurrentItem = NK)
          link ContentToRestOfContent
            ( from ContentData to ContentData
             parameters CurrentItem = 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 a logical notion that can be represented and implemented in several ways and with different syntaxes. This property guarantees the generality of the approach with respect to actual languages and tools used to implement the adaptive application. For instance, we can implement a configuration using SQL at the content level, XHTML at the navigation level and a set of CSS files at the presentation level.

2.3 The basic adaptation methodology

Our process of adaptation is based on the notions of profiles and configurations presented above and on a notion of *matching* between profiles and configurations. Intuitively, a configuration *matches* with a profile if it meets the adaptation requirements of the profile. We will show in the next section how this relationship can be precisely defined.

Assume to have at disposal an initial set of configurations $\mathbf{C}$ that capture the criteria of adaptations for a basic set of profiles. The only requirement is that $\mathbf{C}$ contains at least one configuration that match the generic profiles of a context. Clearly, this set needs to be refined and enriched during the life cycle of the adaptive system.

The adaptation methodology is composed by a number of steps as follows.

1. First, the context of the client is captured and represented in terms of a set of profiles, one for each coordinate of adaptation. Each profile is expressed in the model presented in the previous section. As usual, some aspects of the context can be provided explicitly by the client (e.g., device capabilities), others can be specified implicitly (e.g., user preferences can be derived from the analysis of his/her navigation).

2. For each profile P of the context we select in $\mathbf{C}$ a configuration that matches with P.

3. The set of configurations generated in the previous step are merged into a unique configuration C (e.g., by combining, if possible, the device requirements with the preferences of the users).

4. The configuration C can be further refined to meet the requirements of a specific request done by the user (e.g., an additional data selection);

5. The final configuration C is translated into a corresponding set of *adaptation statements* that implement the configuration in the actual languages for the systems adopted at the various levels. Adaptation statements may correspond to: SQL statements at the content layer, XHTML or JSP statements at the navigation layer, and CCS style sheets at the presentation layer.

6. The adaptation statements are executed by the underlying systems and the final response is generated.

In [5] we have addressed the problem of capturing and translating possibly heterogeneous contexts into a uniform formalism (Step 1 of the above methodology). The rest of this paper focuses on the automatic generation of configurations matching with given profiles and the subsequent management of configurations (the main steps of the methodology).

3 Automatic adaptation

3.1 Adaptation rules

The matching relationship between configurations and profiles is represented by means of a novel notion of *adaptation rule*. An adaptation rule has the form $P_r : C_d \Rightarrow C_f$ where: (i) P_r is a parametric profile, that is, a profile in which parameters can appear in place of values, (ii) C_d is a condition, made of a a conjunction of atoms of the form $A = c$ or $A = B$ where A and B are parameters occurring in P_r and c is a constant value, and (iii) C_f is a parametric 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 then generate the configuration C_f.

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

$$H(\mathit{ImgCapable} : X, \mathit{ScreenSize} : Y, \mathit{ColorCapable} : Z) : X = \mathit{false}, Y < \mathit{1500} \Rightarrow \{q, h, s\}$$

where X, Y, and Z are the parameters of H and $\{q, h, s\}$ is the following parametric configuration:

$$q = \{\ T : x_1, S : x_2, D : x_3, C : x_4 \mid \mathit{NewsItems}(N : x_0, T : x_1, S : x_2, D : x_3, G : x_5, J : x_6), \mathit{Details}(NK : x_0, P : x_7, C : x_4)\}$$

```
h =  IndexUnit NewsIndex
       ( source News Items; attributes Title, Date;
         orderby Date )
     DataUnit ContentData
       ( selector Item = CurrentItem;
         attributes Title, Date, Content)
     link NewsToDetails
       ( from NewsIndex to ContentData
         parameters CurrentItem = NK)
s =  Text( Color: Black, Size: 8pt)
     Link( Style: Underline, Color: Black)
     Image( Size: Y × 0,5, Color: Z)
```

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

A profile P *activates* a rule $P_r : C_d \rightarrow C_f$ if $P_r \lhd P$. Hence, a profile P can activate a rule for a profile that is more general than P.

Now, let R be a rule $P_r : C_d \rightarrow C_f$ activated by a profile P and let C be the configuration obtained from C_f by substituting the parameters of P_r with the actual values occurring in P: we say that C *is generated* by P using R.

3.2 Composition of configurations

Let K be an input context and assume that we have define a set of adaptation rules $\mathbf{R}$. On the basis of the activation mechanism described in the previous section, we can generate a configuration C for each profile P occurring in K, using the rules of $\mathbf{R}$.

Then, according to the adaptation methodology presented in Section 2.3, we need to merge these configurations. To this aim, we introduce a special composition operation $\oplus$ over configurations. Given a pair configurations $C_1(q_1, h_1, s_1)$ and $C_2(q_2, h_2, s_2)$, $C_1 \oplus C_2$ is a configuration $C(q, h, s)$ is defined as follows: (i) $q = q_1 \circ q_2$, that is, q is obtained as the composition of q_1 followed by q_2; (ii) h is obtained by merging h_1 and h_2: if some conflict arises, the choices of h_1 are preferred to those of h_2; and (iii) $s(w_i) = s_1(w_i)$ 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).

Note that the $\oplus$ operation is indeed a *prioritized* composition since if two configurations present conflicts, then the choices done in the configuration on the left hand side of $\oplus$ are preferred to the choices done in the other configuration. This property is used by applying the $\oplus$ operation according to a predefined order of precedence over the adaptation rules.

As an example, assume that $\mathbf{R}$ contains the rule reported in section 3.1 and the following adaptation rule for a user profile U:

$U(\mathit{PreferredGenres} : X, \mathit{PreferredFont} : Y) : \mathit{true} \Rightarrow$
$q = \{\ T : x_1, S : x_2, D : x_3, C : x_4, N : x_5, P : x_6 \mid$
$\quad \mathit{NewsItems}(N : x_0, T : x_1, S : x_2, D : x_3, G : x_7, J : x_8),$
$\quad \mathit{Details}(NK : x_0, P : x_6, C : x_4),$
$\quad \mathit{Newspapers}(J : x_8, N : x_5, C : x_9), x_7 = X\}$
$h = \emptyset$
$s =$ **Text**(**Font**: Y, **Size**: 12pt)

Assume also that the rule for the device takes precedence on the rule for the user. Then, the context $\{H(\mathit{ImgCapable} : \mathit{False}, \mathit{ScreenSize} : \mathit{1000}, \mathit{ColorCapable} : \mathit{false}), U(\mathit{PreferredGenres} : \mathit{Sport}, \mathit{PreferredFont} : \mathit{Arial})\}$ activates these two rules and the combination of the generated configurations produces the following final configuration.

$q = \{\ T : x_1, S : x_2, D : x_3, C : x_4 \mid$
$\quad \mathit{NewsItems}(N : x_0, T : x_1, S : x_2, D : x_3, G : x_5, J : x_6),$
$\quad \mathit{Details}(NK : x_0, P : x_7, C : x_4), x_5 = \mathit{sport}\}$
$h =$ **IndexUnit** NewsIndex
(**source** News Items; **attributes** Title, Date;
orderby Date)
DataUnit ContentData
(**selector** Item = CurrentItem;
attributes Title, Date, Content)
link NewsToDetails
(**from** NewsIndex **to** ContentData
parameters CurrentItem = NK)
$s =$ **Text**(**Font**: Arial, **Color**: Black, **Size**: 8pt)
Link(**Style**: Underline, **Color**: Black)
Image(**Size**: 500, **Color**: Black)

4 Conclusions

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. To guarantee the generality of the approach we have introduced a generic notion of profile, which can be used to model a variety of contexts in a uniform way, and configuration, which can used to describe how to build a suitable adaptation. The adaptation is achieved automatically by means of special 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 effectiveness and flexibility.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] S. Ceri, F. Daniel, V. Demald, and F. M. Facca. An Approach to User-Behavior-Aware Web Applications. In *5th Int. Conf. on Web Engineering (ICWE), Springer-Verlag, Australia*, 2005.

[3] S. Ceri , P. Fraternali, A. Bongio , M. Brambilla, S. Comai, and M. Matera. *Designing Data-Intensive Web Applications.* Morgan Kaufmann, 2003.

[4] R. De Virgilio and R. Torlone. A General Methodology for Context-Aware Data Access. In *4th Int. ACM Workshop on Data Engineering for Wireless and Mobile Access (MOBIDE)*, Baltimore, USA, 2005.

[5] R. De Virgilio and R. Torlone. Management of Heterogeneous Profiles in Context-aware Adaptive Information System. In the *Int. Workshop on Context-Aware Mobile Systems (CAMS)*, Agia Napa, Cyprus, 2005.

[6] Z. Fiala, M. Hinz, K. Meissner, and F. Wehner. A component-based approach for adaptive dynamic Web documents. *Journal of Web Engineering*, 2(1–2):58–73, 2003.

[7] F. Frasincar, P. Barna, G.J. Houben, and Z. Fiala. Adaptation and reuse in designing Web information systems. In *Int. Conference on Information Technology: Coding and Computing (ITCC), IEEE Computer Society, Scotland*, 2004.

[8] F. Frasincar, G. J. Houben, and R. Vdovjak. Specification framework for engineering adaptive Web applications. In *11th Int. World Wide Web Conference (WWW), Web Engineering Track*, 2002.

[9] T. Isakowitz, E. A. Stohr, and P. Balasubramanian. RMM: a methodology for structured hypermedia design. *Communications of the ACM/*, 38(8):34–44, 1995.

[10] R. Vdovjak, F. Fransincar, G.J. Houben, and P. Barna. Engineering semantic Web information systems in Hera. *Journal of Web Engineering*, 2(1–2):3–26, 2003.

[11] W3C. Device Independence Principles. *Internet document: http://www.w3.org/TR/di-princ/*, 2003.