Building Reactive Web Applications

Federico M. Facca Stefano Ceri
Dipartimento di Elettronica e Informazione
Politecnico di Milano
P.za Leonardo da Vinci 32, I-20133 Milano, Italy
{facca,ceri}@elet.polimi.it

Jacopo Armani Vera Demaldé
Istituto di Tecnologie della Comunicazione
Università della Svizzera Italiana
Via Lambertenghi 10 A, CH-6904 Lugano, Swiss
{armanij,demaldev}@lu.unisi.ch

ABSTRACT

The *Adaptive Web* is a new research area addressing the personalization of the Web experience for each user. In this paper we propose a new high-level model for the specification of Web applications that take into account the manner users interact with the application for supplying appropriate contents or gathering profile data. We therefore consider entire *processes* (rather than single *properties*) as smallest information units, allowing for automatic restructuring of application components. For this purpose, a high-level *Event-Condition-Action* (ECA) paradigm is proposed, which enables capturing arbitrary (and timed) clicking behaviors.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia—*architectures, navigation, user issues*

General Terms

Design, Languages, Theory

Keywords

Adaptive Web, Eca Rule, User Modeling, Design Method

1. INTRODUCTION

As the Web is a steadily growing environment and users, rather than navigating relatively simple (static) Web sites with structures that evolve only very slowly in time, nowadays users are more and more faced with complex Web applications, dynamically generated contents and highly variable site structures. Continuously, they are confronted with huge amounts of non pertaining contents or changed interaction paths. As a consequence, users may feel uncomfortable when navigating the Web.

Several techniques have been introduced that aim at augmenting the efficiency of navigation and content delivery from different points of view, among them profile-based *personalization*, *context-aware* or *adaptive* Web applications, *workflow-driven* Web applications, usability studies and Web log analysis efforts.

We believe that a new approach and an open paradigm that combines adaptive and process-centric perspectives can open new ways for both (coase-grained) application adaptation and (online) usability analysis. In this paper we propose a model and a methodology to easily design *behavior-aware* Web applications that allow performing actions in response to the user's fulfillment of predefined navigation patterns. Our proposal is based on the conceptual framework provided by the already established modeling language WebML [1], but we also propose a new formalism, WBM (*Web Behavior Model*) [3], a simple and intuitive model for describing navigation goals. The two models are combined to form a high-level *Event-Condition-Action* paradigm providing the necessary expressive power for capturing the way users interact with applications. Despite the adoption of WebML for hypertext design and WBM for user behaviour modelling, the proposed solution is of general validity and can thus be applied to arbitrary Web applications.

WWW 2005, May 10–14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

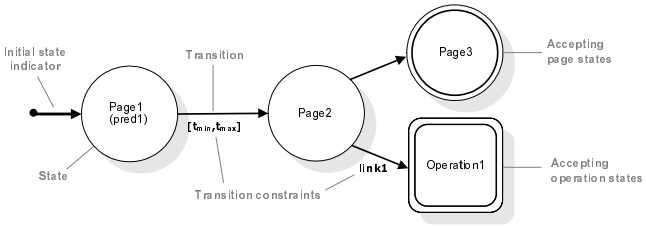

Figure 1: Example of WBM script with state, link, time constraints and multiple exiting transitions from one state. Basic WBM primitives are named: page states are expressed by circles, operation states by rectangles.

2. WBM: AN OVERVIEW

The *Web Behavior Model* (WBM) is a timed state-transition automata for representing classes of user behaviors on the Web. Graphically, WBM models are expressed by labeled graphs, allowing for an easily comprehensible syntax; cf. Figure 1. A *state* represents the user's inspection of a specific portion of Web hypertext (i.e., a page or a collection of pages), which is loaded on his browser, or the user's activation of a specific Web operation, such as "buy" on an e-commerce site, or "download" of a given file. A *transition* represents the navigation from one state to another.

Figure 1 shows an example WBM script. Entering the state denoted by *Page1* is constrained by *pred1*, which must evaluate to true. The transition from the first state to the second state must occur within t_{min} and t_{max} time units

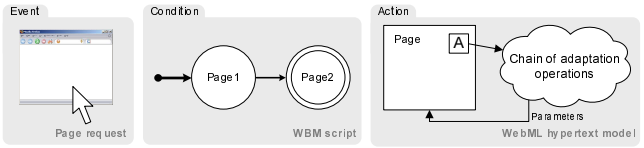

Figure 2: High-level ECA rule components.

from the moment the script has been initiated, otherwise the script fails. The script in Figure 1 also presents two exiting transitions from state *Page2*. States labeled *Operation1* and *Page3* are "competing", as a browsing activity in *Page2* may lead to either *Operation1* or *Page3*. Therefore, either a user browses from *Page2* to *Page3* (the transition *Page2* to *Page3* is triggered) and the script reaches the accepting state denoted as *Page3*, or the transition to accepting state *Operation1* occurs if *Operation1* is performed by using *link1*. WBM is inspired by WebML, especially what regards constraint definitions; however, its core components can also be used to describe arbitrarily developed hypertexts.

3. ECA RULE MODEL

To build the ECA rules that finally make Web applications aware of predefined user behaviors, we now combine WBM scripts and WebML adaptation mechanisms [2]. Commonly, the ECA paradigm describes a general syntax: `on` *event* `if` *condition* `do` *action*. In our view, the event consists in a *page request*, the condition is a set of requirements on the user navigations (expressed as a *WBM script*), and the action part specifies some adaptivity actions to be forced in the Web Application and expressed as *WebML operation chain*. Events are generated only for explicitly labeled pages (`A`) denoting the *scope* of the rule, and proper rule priorities resolve possible conflicts among concurrently activated rules.

Consider for instance the rule of Figure 2. The rule reacts to a user's visit to *Page1* followed by a visit to *Page2*. Thus, the expressed condition only holds when the script gets to the accepting state *Page2*. Once the accepting state is reached, the actions (expressed as cloud) are executed and, after a re-computation of page parameters, possible adaptations may be performed.

4. ARCHITECTURE

Our ReActive Web framework is illustrated in Figure 3. The Rule Engine collects and evaluates HTTP requests in order to track a user's navigational behavior, and hosts a repository of WBM scripts, which can be executed on behalf of individual users.

The behavior of the Rule engine is described by the following steps: (i) URL requests as generated by user clicks are notified to the Rule Engine; (ii) a request can cause either the instantiation of a new script, or a state change of a given running script, or nothing. (iii) When a WBM script reaches an accepting state for a certain user, the Rule Engine changes a record in the shared database, storing the information about the completed script and the user's session. Also, variables used by the WBM script are stored in the database. (iv) Finally, if the request refers to a page contained within the rule's scope, the application interprets the modified data record as request for activating the adaptation

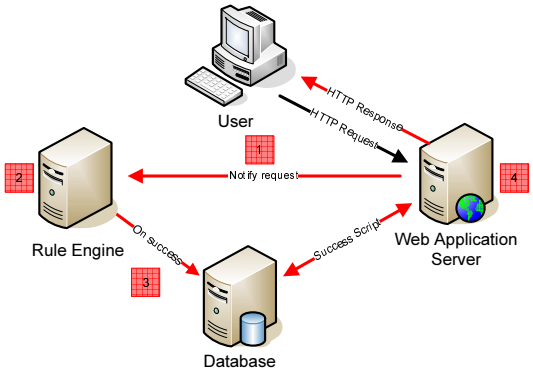

Figure 3: A schema representing the architecture of the ReActive Web System.

chain associated to that page. Accordingly, it executes the operation chain, possibly generating a modified hypertext.

Synchronous as well as *asynchronous* rule execution models can be achieved with the presented architecture. In the synchronous case, if a rule is successfully triggered, the action part of the rule is executed immediately at the first page request. To avoid possible performance slowdowns (due to time spent for script evaluation), the asynchronous configuration defers rule evaluation to the next (automatic) page refresh. This allows for task parallelization and short response times. The strong decoupling of application server and Rule Engine allows for independent resource management and parallelized and scalable configurations.

5. CONCLUSIONS

In this poster we proposed a general purpose model for building behavior-aware Web applications. Our proposal is based upon WebML and WBM, and combines these two models into a visual ECA paradigm that opens the road to the implementation of high-level CASE tools for designing advanced websites. A first prototype of the presented architecture has been developed and tested by implementing an e-learning ReActive Web Application. First feedback from experiments are quite positive: experiments proved that the whole mechanism is feasible and that the use of the asynchronous execution model effectively avoids Rule Engine response times to impact on user navigation.

A complete example of the ReActive Web approach is presented on the full-size poster.

6. REFERENCES

[1] Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M.: Designing Data-Intensive Web Applications. Morgan Kauffmann (2002)

[2] Ceri, S., Daniel, F., Matera, M.: Extending webml for modeling multi-channel context-aware web applications. In: Proceedings of Fourth International Conference on Web Information Systems Engineering Workshops (WISEW'03), Rome, Italy, December 12 -13, 2003, IEEE Press (2003) 225–233

[3] Armani, J., Ceri, S., Demaldè, V.: Modeling of user's behaviors in web applications: a model and a case study. Technical report, Institute of Communication Technologies, Università della Svizzera italiana (2004)