Context-aware Provision of Advanced Internet Services *

Raffaella Grieco Delfina Malandrino
Università degli Studi di Salerno, Italy
Dipartimento di Informatica ed Applicazioni
"R.M. Capocelli"
84081 Baronissi (SA), Italy
{delmal, rafgri}@dia.unisa.it

Francesca Mazzoni
Università degli Studi di Modena
e Reggio Emilia, Italy
Dipartimento di Ingegneria dell'Informazione
41100 Modena, Italy
mazzoni.francesca@unimore.it

Daniele Riboni
Università degli Studi di Milano, Italy
Dipartimento di Informatica e Comunicazione
20135 Milano, Italy
riboni@dico.unimi.it

Abstract

The pervasive and mobile computing scenario is characterized by the heterogeneity of devices used to access services, and by frequent changes in the user's context. This paper presents the integration of two existing frameworks into a context-aware content provisioning system oriented to mobile users. External Web contents can be automatically tailored according to a wide range of context data, including device capabilities, available bandwidth, location, user preferences. The tailoring process is human intervention free. In order to demonstrate the feasibility of our solution, we have developed a prototype location-based service that takes advantage of this architecture.

1 Introduction

In the *Pervasive and Ubiquitous Computing* era, the trend is to provide Web content and multimedia applications by taking into account four important requirements: *anytime-anywhere* access to *any data* through *any device* and by using *any access network*. As a consequence, it is increasingly significant to provide tailored content efficiently, in order to address the mismatch between the rich multimedia content that is widely available and the limited client capabilities and available bandwidth. Furthermore, users feel the need of personalized content that matches their personal preferences, considering not only device capabilities and network status, but a wider notion of *context*, which includes -among other things- their current location, activity, and surrounding environment.

One of the current research topics in distributed systems is how to extend the traditional client/server computational paradigm in order to allow the provision of *intelligent* and *advanced* services. To this aim, new actors are introduced into the WWW scene, in the form of intermediaries that act on the HTTP data flow exchanged between client and server. Several examples of intermediary adaptation systems exist in literature, such as iMobile by AT&T [7], whose main goal is to provide personalized mobile services; the TACC infrastructure [4], whose main goal is to support the dynamic deployment of transformation modules into proxy-based components; RabbIT[1] and Privoxy[2], whose main goal is to provide functionalities such as text and image transcoding, removing of cookies, GIF animations, advertisement, Java Applets, etc. Many of these systems provide adaptation functionalities without taking into account user's preferences and context information.

In order to support the provisioning of this type of services, we have defined and experimented with an architecture that builds on top of two existing frameworks: the *Scalable Intermediary Software Infrastructure (SISI)* [3], and the *Context Aggregation and REasoning (CARE)* [1] middleware. Our proposal is supported by a running implementation, and by a prototype service aimed at providing location-based support to mobile users.

The rest of the paper is organized as follows. Section 2

*This work has been partially supported by Italian MIUR (FIRB "Web-Minds" project N. RBNE01WEJT_005).

[1] http://rabbit-proxy.sourceforge.net
[2] http://www.privoxy.org

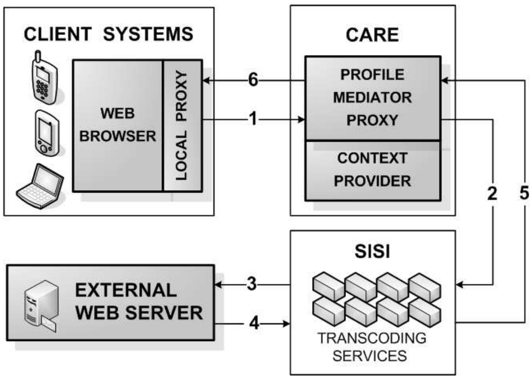

Figure 1. The adaptation architecture.

illustrates the architecture and the data flow; Section 3 describes the dynamic adaptation mechanism; Section 4 describes the prototype service; Section 5 deals with future work.

2 Architecture overview

In this section we present the architecture for context-aware service provisioning.

Figure 1 shows the system architecture and the data flow upon a user request. In order to provide the system with the user's identification and with the information required to retrieve distributed profiles and policies, the request is intercepted by a LOCAL PROXY that runs on the client device and adds these data into custom HTTP headers. Then (Step 1), the request is forwarded to the node running the CARE middleware for context-awareness. Here the PROFILE MEDIATOR PROXY retrieves the aggregated context data from the CONTEXT PROVIDER module. These data -which include the list of SISI services to be applied, as well as their parameters- are inserted into the HTTP headers, and the request is sent to the node running the SISI adaptation system (Step 2). SISI retrieves the requested resource from the external Web server (Steps 3 and 4), and applies the services on the basis the user's context data and preferences. Finally, the adapted resource is sent back to the user's client (Steps 5 and 6) trough the CARE node.

In the following of this section we briefly describe the two main components of the presented architecture: the *CARE* middleware for context-awareness and the *SISI* adaptation system.

2.1 The CARE middleware for context-awareness

CARE, presented in detail in [1], aims at supporting context-aware mobile services in an environment characterized by distributed context sources. CARE has been designed in order to support a wide range of technologies and application servers. In the described architecture, CARE is in charge of providing SISI with those context data that are useful to determine the activation and the parameters of services as well as users' preferences.

We call *profile* a subset of context information collected and managed by a certain entity. Profiles are represented by using the CC/PP language [6], and are managed by three entities, namely: the user with his/her devices; the network operator with its infrastructure; and the service provider. Each entity has a dedicated profile manager to handle its own context data. The middleware includes ontology services for managing and reasoning with socio-cultural context data that cannot be modeled in CC/PP. Adaptation and personalization parameters are determined, at the time of the service request, by policy rules defined by both the user and the service provider, and managed by their corresponding profile managers. The CONTEXT PROVIDER module is in charge of calculating the aggregated context information that will be used by the application logic for the adaptation services. In particular, it retrieves context data from the profile managers, and evaluates adaptation policies solving possible conflicts arising among context data and/or policies. The ad-hoc rule-based reasoning services of the CONTEXT PROVIDER are particularly efficient. Experimental results [2] have shown that the evaluation of policy rules is executed in few milliseconds.

2.2 The SISI adaptation system

The main goal of SISI [3, 5] is to facilitate the deployment of efficient and programmable adaptation services.

This framework, built on top of Apache Web server and mod_perl, provides a modular architecture that allows an easy definition of new functionalities implemented as building blocks in Perl. These building blocks, packaged into *Plugins*, produce transformations on the information stream as it flows through them. Moreover, they can be combined in order to provide complex functionalities. Thus, multiple Plugins can be composed into SISI edge services, and their composition is based on preferences specified by end users.

We already implemented many adaptation services, among which we can mention:

LinkRelationship. The HTML *LINK* element can be used to improve the Web site accessibility and, at the same time, to ensure a better support for search engines (e.g. semantic information). Our service adds a toolbar containing the LINK attributes on top of each HTML page. The main goal is to make HTML pages more accessible when "read" through screen readers, and also to improve their navigability on devices with limited capabilities.

BlockList and AnnoyanceFilter. The main goal of these services is to get rid of particularly annoying abuse during the navigation on the Web. In particular, the BlockList service provides a simple way to block a Web browser from viewing sites that are not on a list of approved sites. This service searches for all the links embedded in a Web page and substitutes the unapproved ones by plain text.

The AnnoyanceFilter service provides functionalities for removing advertisement, banners, pop-ups in JavaScript and HTML, JavaScript code, for disabling unsolicited pop-up windows, etc.

3 Service activation policies

By default, each SISI service is deactivated. The activation of services, as well as the service parameters, are determined by policy rules declared by both the user and the service provider. Policies are essentially conditions on context data that set the value of other profile attributes when satisfied.

Example 1 *Consider the case of the* FilterImg *service for image transcoding. The evaluation of the following policy rules determine the activation state and the parameters of the service:*

R1: ***If*** DeviceType = *CellPhone*
Then Set FilterImg:Activate=*'on'*

R2: ***If*** DeviceType = *CellPhone* ***And*** Bearer ! = *UMTS*
Then Set FilterImg:removeImage=*'on'*

R3: ***If*** ColorCapable = *no*
Then Set FilterImg:black&white=*'on'*

Rule R1 is used to activate the service for devices with low capabilities (cell phones). Rule R2 instructs SISI to remove images when the cell phone is not connected through a UMTS bearer. Rule R3 is used to determine the transformation in black and white of images for devices that cannot display colors. It should be noted that the user's preferences regarding the activation of services, as well as their parameters, are represented through a proper CC/PP vocabulary.

Since policies can be declared by both the user and the service provider, conflicts can arise between policies setting a value for the same service parameter. When conflicting policies are declared by the same entity, the conflict is easily solved by allowing the entity to specify an explicit priority between conflicting rules. On the other hand, when a conflict is encountered between a policy declared by the user and a policy declared by the service provider, this type of conflict is resolved on the basis of *resolution directives* declared by the service provider (see [2] for more details). With regard to the adaptation services, our choice was to give the highest priority to the user, while letting the service provider to specify default policies.

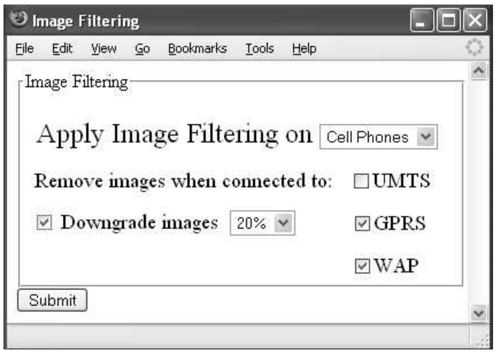

Figure 2. User interface to declare policies.

Example 2 *Consider the case of the* FilterImg *service, and suppose that the rules of Example 1 are the default rules declared by the service provider. Suppose that a particular user declares the following policy rule to disable the removal of images when the cell phone is connected through GPRS:*

R4: ***If*** DeviceType = *CellPhone* ***And*** Bearer = *GPRS*
Then Set FilterImg:removeImage=*'off'*

When a cell phone is connected through GPRS, both rules R2 and R4 could fire, determining conflicting values (i.e., 'on' and 'off') for the same parameter. In this case, since our choice was to give higher priority to the user's policies, rule R4 is evaluated first, and sets the value 'off' to the FilterImg:removeImage *parameter. Since rule R4 fired, rule R2 is discarded.*

Of course, policies expressed by means of the formal language shown in Examples 1 and 2 are not intuitive for the final users of the system. For this reason, users are provided with Web interfaces to manage their preferences regarding the activation and parameters of SISI services. Figure 2 shows the interface to express preferences regarding the *FilterImg* service.

4 A prototype service

In order to show the integration between CARE and SISI we have implemented a prototype location-based service, named *GeoAware*. This service is addressed to mobile users equipped with a mobile device and a GPS receiver. The main goal of this service is to provide a map with information about both the current location of the user and the locations (expressed as physical addresses) appearing on the Web page she is currently viewing. The main steps performed by the GeoAware service are described below.

At first, CARE retrieves the GPS coordinates of the user from her profile manager, and communicates them to SISI

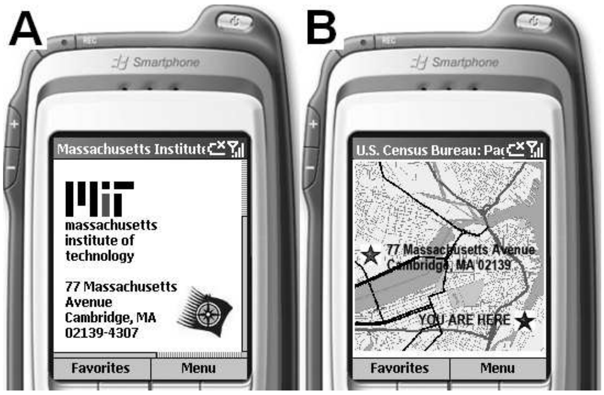

Figure 3. The GeoAware service.

as explained in Section 2.

The GeoAware service parses the requested Web page and, by applying regular expressions, matches all standard U.S. addresses [8]. When an address is recognized, GeoAware adds a hyperlink on the Web page, highlighted by a particular icon (see Figure 3-A).

When the user selects the hyperlink icon, GeoAware invokes the *geocoder.us* service[3] to obtain the coordinates of the relative address. Geocoder.us is a public service providing free geocoding of addresses in the United States, and relies on Geo::Coder::US, an open-source Perl module available from CPAN.

After having obtained this information, GeoAware builds a query string to be issued to the U.S. Census Bureau *TIGER Map Server*[4], which provides public-domain, customized U.S. maps. Figure 3-B shows the map obtained from the address in Figure 3-A. The current position of the user is represented by a blue star with a label YOU ARE HERE, while the position corresponding to the address is represented by a red star labeled with the address. Exploiting the other adaptation services, the map is properly tailored to device capabilities and available bandwidth.

The interest for geolocalized services is witnessed by the recent service provided by Google with its AutoLink button in the Google toolbar. Even if that service addresses the same needs as ours, it is currently available only to users with desktop browsers. Moreover, the AutoLink functionalities are somewhat limited since it does not take advantage of a context-awareness framework.

5 Future Work

With regard to scalability issues, we plan to distribute the adaptation process among different nodes in a clustered environment, in order to support a very high number of concurrent requests. Currently, we are considering to replicate the adaptation servers and to study some policies for obtaining a good load sharing.

Then we will try to speed up the adaptation process, partially counterbalancing the overhead due to distribution by adding some caching mechanisms. This approach should reduce the server load, as well as the user perceived latency during Web navigation.

Finally, we are also defining and implementing some optimizations for reducing unnecessary computations on the context-awareness framework. A simple optimization is to avoid the re-computation of policies when context data do not change during the user session. We are also investigating some more complex approaches based on caching and triggers.

[3] `http://geocoder.us/`

[4] `http://tiger.census.gov/cgi-bin/mapbrowse-tbl`

References

[1] A. Agostini, C. Bettini, N. Cesa-Bianchi, D. Maggiorini, D. Riboni, M. Ruberl, C. Sala, and D. Vitali. Towards Highly Adaptive Services for Mobile Computing. In *Proceedings of IFIP TC8 Working Conference on Mobile Information Systems (MOBIS)*, pages 121–134. Springer, September 2004.

[2] C. Bettini and D. Riboni. Profile Aggregation and Policy Evaluation for Adaptive Internet Services. In *Proceedings of The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous)*, pages 290–298. IEEE Computer Society, 2004.

[3] M. Colajanni, R. Grieco, D. Malandrino, F. Mazzoni, and V. Scarano. A Scalable Framework for the Support of Advanced Edge Services. In *Proc. of the 2005 Int. Conf. on High Performance Computing and Communications (HPCC'05)*, pages 1033–1042, Sorrento (NA), Italy, September 2005.

[4] A. Fox, Y. Chawathe, and E. A. Brewer. Adapting to Network and Client variation using active proxies: Lessons and perspectives. *IEEE Personal Communications*, 5(4):10–19, 1998.

[5] R. Grieco, D. Malandrino, F. Mazzoni, and V. Scarano. Mobile Web Services via Programmable Proxies. In *Proc. of the IFIP TC8 Working Conference on Mobile Information Systems - 2005 (MOBIS)*, pages 139–146, Leeds (UK), December 2005.

[6] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler, and L. Tran. Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation, W3C, January 2004. http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/.

[7] C. Rao, Y. Chen, D.-F. Chang, and M.-F. Chen. imobile: A proxy-based platform for mobile services. In *Proceedings of the First ACM Workshop on Wireless Mobile Internet (WMI 2001)*. ACM Press, 2001.

[8] US Postal Service. Postal Addressing Standards. Technical Report Publication 28, November 2000.