MOBILE-WEB SERVICES VIA PROGRAMMABLE PROXIES*

Raffaella Grieco[1], Delfina Malandrino[1], Francesca Mazzoni[2], Vittorio Scarano[1]

[1]*ISISLab, Dipartimento di Informatica ed Applicazioni "R.M. Capocelli"*
Università di Salerno, 84081 Baronissi (Salerno), Italy
{rafgri, delmal, vitsca}@dia.unisa.it

[2]*Dipartimento di Ingegneria dell'Informazione*
Università di Modena e Reggio Emilia, 41100 Modena, Italy
mazzoni.francesca@unimo.it

Abstract Our goal, in this paper, is to present the effectiveness of an intermediary framework to provide mobile-oriented services via edge services. To this end we developed services for device independence in such a way that content is adapted according to the capabilities of the target devices.

Keywords: Programmable proxies, device independence, ubiquitous Web.

1. Introduction

The World Wide Web is, nowadays, a ubiquitous infrastructure for any information and communication technology. In fact, it is globally recognized as a universal framework to *anytime-anywhere* access *any data* or information from any networked computer or device and by using *any access network*.

The proliferation of the rich, entertaining and interactive applications available on the Web is mimicked by the comparable growth in wireless technologies that spawn a variety of terminal devices such as Desktop PC, pagers, personal digital assistants (PDAs), web-phones, etc.

Of course, all the devices target different needs and users' types and, therefore, offer an equally variegated range of characteristics such as storage, display capabilities (such as screen size and color depth), wireless network connections, limited bandwidth, processing power and power consumption. These

*This research work was financially supported by the Italian FIRB 2001 project number RBNE01WEJT "WEB–MiNDS" (Wide-scalE, Broadband MIddleware for Network Distributed Services).

constraints currently pose several challenges on the designer since both the delivery and the presentation of complex personalized applications must take into account the limitations of the device.

The key to meet the demands in this heterogeneous environment is the adaptation of contents to the capabilities of the devices and communication systems.

In this paper, we present content adaptation services for the Mobile Web as implemented on a programmable proxy infrastructure called Scalable Intermediary Software Infrastructure (SISI). In particular, we present the implementation on top of our SISI framework of Content Selection markup language [1] that was recently released (May 2^{nd}, 2005) by the World Wide Web Consortium (W3C) Device Independence WG (DIWG) [http://www.w3.org/2001/di/].

2. Proxy Frameworks for mobile computing

One of the current research trend in distributed systems is how to extend the traditional client/server computational paradigm in order to allow the provision of *intelligent* and *advanced* services.

This computational paradigm introduces new actors within the WWW scene, the intermediaries [2, 3], i.e. software entities that act on the HTTP data flow exchanged between client and server by allowing content adaptation and other complex functionalities, such as geographical localization, group navigation and awareness for social navigation, translation services, adaptive compression and format transcoding, etc.

Several popular proxy systems, such as RabbIT [http://rabbit-proxy.sourceforge.net/], WebCleaner [http://webcleaner.sourceforge.net] and Privoxy [http://www.privoxy.org/] provide functionalities for text and image transcoding, removing cookie and GIF animations, removing advertisement, banner, Java Applets and JavaScripts code, protecting privacy and controlling access.

Web Based Intermediaries (WBI) [http://www.almaden.ibm.com/cs/wbi/] is a programmable proxy whose main goal is to simplify the development and the deployment of Web intermediary applications (i.e. applications that deal with HTTP information flows) and in particular content adaptation and personalization.

AT&T Mobile Network [4] is a proxy-based platform designed to provide personalized services to users of mobile devices. It provides a modular architecture that allows an easy definition of new functionalities on top of a programmable proxy. The modularity is achieved through three different abstractions: devlets that provide protocol interfaces to different mobile devices; infolets that collect information from different data sources and applets that encapsulate the service's functionalities.

MARCH [5] is a distributed content adaptation architecture that provides functionalities for adapting Web content according to the capabilities of client devices that access to Internet services. An important feature of this architecture is that it allows the dynamic composition of services for a given set of operating conditions (CPU, memory, etc).

3. Scalable Intermediary Software Infrastructure (SISI)

In this section we provide a brief description of the SISI architecture that we are currently developing and provide an overview of its implementation and a brief description of the mechanisms we used to personalize the services (for each user).

Our framework is based on top of existing open-source, mainstream applications, such as Apache Web server and mod_perl, because, first of all, of the quality of these products of open-source development and, secondly, because the results will be widely usable given the popularity of the Apache Web server.

SISI framework focuses on providing a simple approach for assembling and configuring complex and distributed applications from simple basic components. The idea is to provide functionalities that allow programmers to develop services without taking care of the details of the infrastructure that will host these services. Therefore, SISI provides a modular architecture that allows an easy definition of new functionalities implemented as building blocks in Perl. These building blocks produce transformation on the information stream as it flows through them. For a detailed description of SISI architecture and its implementation please refer to [6, 7].

Users' profiles configuration. User and device profiling is a very important characteristic because of the heterogeneity of the devices and services of the Mobile Web. Users need multiple profiles according to their status or preferences, and, then, different configurations that must be easily switched to as well as modified. Of course, configuration must be performed by the user itself.

Our approach in managing user profiles is to explicit ask the user what s/he needs and use this information with a rule-based approach to personalize the content. In particular, users have to fill-out forms to create new profiles[1] and to modify or delete the existing ones.

Furthermore, because different client capabilities of the devices used to access the Web, users must be allowed to specify, in their device's profiles, the needed parameters both for devices and services. For example, when a user connects with a PDA s/he could want to be displayed only black and white im-

[1] The term profile, in this paper, is referred to SISI profile, i.e., the set of services that are configured (with their own parameters) by the end user. When referring to other kind of profiles, we will use the full definition, like CC/PP profiles, for instance, in Section 4.

ages or not given images at all in order to save bandwidth. To this aim the user only has to activate the corresponding profile previously saved into the system.

SISI allows a simple configuration mechanism for adding, removing, enabling or disabling functionalities and a more complete configuration mechanism where service parameters can be specified by asking the user to fill-out Web-based forms (for example, by providing a downgrade quality image for a transcoding service).

4. Content Adaptation Services

Adapting services to capabilities and preferences of different users and devices is a research issue addressed in both the mobile and the universal access research communities. The mobile community has mostly been concerned with device capabilities and technical problems, and has thus been more oriented toward solutions with a common presentation for all devices. Then, we first describe our implementation of the Working Draft [1] of the Device Independence Working Group of the W3C and next we show how SISI profiles interact with the CC/PP profiles.

Content Adaptation Independence Model. The W3C is currently working on activities relevant to content adaptation within the Device Independence Working Group. This Group is studying techniques for content adaptation, authoring and presentation techniques for Web content in order to match the different capabilities of client devices to which contents have to be delivered to.

In order to make the Web accessible anytime and anyhow, in particular by supporting other than many access mechanisms also many modes of use (including visual and auditory ones), they provide some important suggestions or principles as described in [8]. Finally, this Working Group has produced documents for CC/PP recommendation, and documents that discuss authoring challenges and techniques [9]. From these documents it is possible to achieve information about authoring-related principles, and in particular out interest is mainly devoted to the need of, whenever possible, reusing content across multiple delivery context, adapting presentations independently from the access mechanisms.

Finally, a draft on Content Selection markup language [1] discuss how to express alternate contents or resources for their delivering to end users, by changing the presentation according to the capabilities of the requesting device.

Device Independence. The draft on Content Selection [1] specifies a syntax and a processing model for general purpose content selection or filtering. Selection involves conditional processing of an XML information according to the results of the evaluation of expressions. Using this mechanism some parts of the information can be selected while other not delivered, automatically adapting the original content according to particular accessibility rules.

In the SISI Content Selection service (SISI CS service) we are implementing the specifications of the draft. We have currently implemented the conditional expressions, and the conditional expressions that return values, that can be further used to check if a particular piece of content is to be included for processing in the Web page delivered to end users (see Figs. 1 and 2). The remaining part of the specification is currently under testing or development.

If the end user aims to use the CS service, s/he has to configure service's parameters in order to meet the capabilities of the accessing terminal device. If the used device is a mobile one with a small display, the user can accordingly set height and width parameters, specifying the number of pixels that the device is able to support.

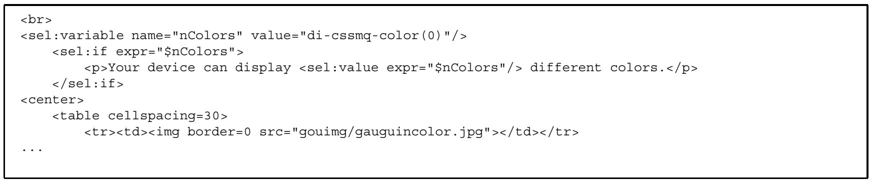
The CS service intercepts user's request, reads the user's profile and the service user-defined parameters, retrieves the Web page from the Web server (the original Web pages augmented with the `DISelect` markup expressions), performs some computation and, finally, delivers only the content that satisfy the specified conditional expressions and rules. Computations include the parsing of the HTML Web page to pull out the `DISelect` tags, invocation of the functions according the matched expressions to validate the values of the corresponding variables.

```
<br>
<sel:variable name="nColors" value="di-cssmq-monochrome()"/>
    <sel:if expr="$nColors">
        <p>Your device can display monochrome images.</p>
    </sel:if>
<center>
    <table cellspacing=30>
    <tr><td><img border=0 src="gouimg/gauguinmonochrome.jpg"></td></tr>
</center> ...
```

Figure 1. An example of conditional expressions.

The function `di-cssmq-color(0)`, in the example shown in Fig. 2, returns the number of colors supported by the accessing device. If the value is not zero, the paragraph is included in the delivered content by displaying the value of the `nColors` variable in place of the `sel:value` element. The presentation of the Web content will change accordingly to the value of `nColors`. For example, if the user uses a device that supports only color depth 256, then the CS service will retrieve only the embedded images with the specified resolution. Of course, the embedded images are available on the Web server at different color resolution, but only that one that matches the conditional expression is provided and displayed.

Device Independence and CC/PP. The configuration of SISI CS service consists in specifying the parameters in order to meet the capabilities of the



```
<br>
<sel:variable name="nColors" value="di-cssmq-color(0)"/>
    <sel:if expr="$nColors">
        <p>Your device can display <sel:value expr="$nColors"/> different colors.</p>
    </sel:if>
<center>
    <table cellspacing=30>
        <tr><td><img border=0 src="gouimg/gauguincolor.jpg"></td></tr>
...
```

Figure 2. An example of conditional expressions that return values.

accessing terminal device. The W3C specifications of CC/PP (Composite Capabilities/Preferences Profiles) are used to specify the parameters. A CC/PP profile includes information on the device capabilities (e.g hardware and software characteristics, operating system, etc) and some user preferences.

In practice, a CC/PP profile contains a number of CC/PP attribute names and associated values that are used by a server to determine and deliver the most appropriate content to the client device. A set of CC/PP attribute names, permissible values and associated meanings constitute a CC/PP vocabulary. It is possible that metadata (user preferences and device capacity) are created in different time and resources can be in different place (i.e. vendor profile for a specific device).

To aggregate all the attributes there are the remote (or indirect) references: each reference links to a collection of default characteristics (specially useful for devices with limited bandwidth). This default characteristics are, then, integrated with the values specified by the user in order to realize the current profile and are delivered to the content server using an exchange protocol. The HTTP Extension Framework is a generic extension mechanism for HTTP/1.1 designed to interoperate with the existing HTTP applications and is the proposed CC/PP exchange protocol. The W3C Working Draft [10] talks about the requirements, assumptions and goals of the CC/PP framework.

In our implementation, the set of rules to determine the server behavior on the value of the profile components is represented by the DISelect markup expressions that are evaluated accordingly to the DISelect variables populated by CC/PP profiles. Within SISI, the CC/PP profile can be used in different contexts: a remote CC/PP profile can fill in the values for the (CC/PP) attributes to be used by Content Selection service but can also be (locally) stored for configuration parameters of other services.

Currently there are not many implementations that can support the CC/PP protocol. Examples of implementations are DELI (Open Source Delivery Context Java Library supporting CC/PP and UAProf) [11], PANDA - Skunk (An Open source CC/PP test bed, plug-in for MS's IE and integration CC/PP with P3P) [12], and an implementation on Apache Web Server [13] using a proxy server.

SISI for the Mobile Web. From the architectural point of view, the programmability and users' profile management make SISI an interesting tool for the Mobile Web since it fits well in the proxy scenarios described in [10, 13].

In the first scenario, two proxies are used in the connection, a client proxy and a server proxy, to exchange CC/PP information and then to implement the protocol. SISI can easily manipulate request/response entities, and, therefore, can be placed either as client proxy or server proxy (or both) with two different modules (CC/PP aware) in the HTTP life-cycle.

The second scenario requires includes a server proxy and a client with browser CC/PP enabled (e.g., Panda [14, 15] from the Keio University) that implements transformations from HTTP to HTTP + CC/PP incorporating the client proxy functionality. SISI can be easily used as a standard HTTP proxy, to read the modified HTTP GET which carries the profile or the profile difference. The information extracted is used to retrieve the right content from the Web server or to adapt the response received to the access device capabilities. In case the profile is an inline reference to a profile repository, SISI can be easily programmed to cache profiles on behalf of the client.

The third approach is to implement a CC/PP aware Web server communicating with an intermediary client proxy. SISI can support the functionality of the client proxy, that is, to convert the standard HTTP headers sent by the Web client to extended HTTP headers that contain CC/PP headers (HTTP headers + CC/PP headers).

The last is the ideal approach where both server and client are capable of sending and receiving CC/PP, then, if we want use our framework, it should be a Web server CC/PP aware.

From the previous argument it turns out that SISI framework can be designed to support CC/PP and bridge the gap between the CC/PP-aware device and the server's delivery of device-appropriate content.

5. Conclusions

Without a full support toward the development of new services, advanced mechanisms cannot be easily realized, deployed and managed, while quick prototyping and manageability represent crucial requirements to assure that programmers can quickly respond to mutations of data format, content or standards that are so common in the Web, nowadays.

Therefore, our objective, in this paper, was to show how our intermediary and programmable framework can be used to quickly and efficiently allow the deployment of services that can make navigation on the WWW from mobile terminals an enjoyable and non frustrating experience.

As a final consideration, we believe that SISI profiles are a key factor in providing an effective personal environment where each user can create/modify/delete profiles as well as easily switch among profiles as s/he needs. The integration of SISI profiles (that contains the proxy services activated for each HTTP request) with the CC/PP profiles is particularly fruitful, especially if seen in the context of the results presented in [16] where an architecture for profiles and policy management is presented.

References

[1] DIWG. W3C Working Draft: Content Selection for Device Independence (DISelect) 1.0, 2005. http://www.w3.org/TR/2005/WD-cselection-20050502/.

[2] R. Barrett and P. P. Maglio. Intermediaries: An approach to manipulating information streams. *IBM Systems Journal*, 38(4):629–641, 1999.

[3] A. Luotonen and K. Altis. World-Wide Web proxies. *Computer Networks and ISDN Systems*, 27(2):147–154, 1994.

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

[5] S. Ardon, P. Gunningberg, B. LandFeldt, Y. Ismailov, M. Portmann, and A. Seneviratne. MARCH: a distributed content adaptation architecture. *Intl Journal of Communication Systems, Special Issue: Wireless Access to the Global Internet: Mobile Radio Networks and Satellite Systems.*, 16(1), 2003.

[6] R. Grieco, D. Malandrino, F. Mazzoni, V. Scarano, and F. Varriale. An Intermediary Software Infrastructure for Edge Services. In *Proc. of The 1st Intl. Workshop on Services and Infrastructure for the Ubiquitous and Mobile Internet (SIUMI'05)*, June 2005.

[7] 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 Intl Conf. on High Performance Computing and Communications (HPCC-05)*, September 2005.

[8] DIWG. Device independence principles, 2003. http://www.w3.org/TR/di-princ/.

[9] DIWG. Authoring techniques for device independence. http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/.

[10] CC/PP Working Group. Composite Capabilities/Preferences Profiles: Requirement and Architecture. http://www.w3.org/TR/CCPP-ra/.

[11] Mark H. Butler. Open Source Delivery Context Java Library supporting CC/PP and UAProf. http://www.hpl.hp.com/personal/marbut/DeliUserGuideWEB.htm.

[12] LSB Research. Research Project on Location Based Web Service. http://romeo.slab.sfc.keio.ac.jp/.

[13] C. Papachristos and E. Markatos. A CC/PP aware Apache Web Server. http://dcs.ics.forth.gr/Activities/papers/ccpp_position.pdf.

[14] K. Yasuda. Implementation and Evaluation of Keio CC/PP Implementation. http://yax.tom.sfc.keio.ac.jp/panda/.

[15] K. Yasuda, T. Asada, and T. Hagino. Effects and Performance of Content Negotiation Based on CC/PP. In *Proc. of The Second Intl Conf. on Mobile Data Management (MDM '01)*, pages 53–64, London, UK, 2001. Springer-Verlag.

[16] C. Bettini and D. Riboni. Profile aggregation and policy evalulation for adaptive internet service. In *Proc. of The First Annual Intl Conf. on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous 2004)*, 2004.