Cross Modality Adaptation of Service Front Ends

by Fabio Paternò, Christian Sisti and Lucio Davide Spano

The current evolution of pervasive technologies means that traditional access modalities are not always suitable for a particular service and, consequently, need to be reassessed. There are many scenarios in which the user is unable to use a screen or keyboard to interact with a given service. In such cases, a vocal interaction can be useful, but this is only supported by a few application providers. We propose a solution to adapt existing Service Front Ends in order to support vocal interaction, which is based on reverse engineering, adaptation heuristics and automatic generation of the vocal implementation.

Pervasive environments imply the possibility of accessing interactive applications anywhere and anytime. There are times, however, when a user may need to access a service but is involved in tasks that occupy his or her hands, making it impossible to use a screen or keyboard. The vocal modality is a good candidate for service access in these situations, and this is confirmed by the recent effort dedicated to supporting it by the main players in mobile OS (eg iOS and Android).

Though supporting vocal interaction can be a good opportunity for service providers, the creation of a Service Front End (SFE) for an additional modality requires an effort that not all application developers can afford. Our approach addresses this issue by supporting the automatic adaptation of existing Web SFEs to the vocal modality.

Our solution is based on an adaptation server, which is able to produce a VoiceXML SFE taking as input its HTML+CSS implementation. The final result adapts the hierarchical information structure of the initial SFE, by including menus that facilitate vocal navigation of the content and remove elements not relevant for the vocal rendering. Both input and output techniques are also adapted so that they can be supported by the vocal modality. In this process we exploit the MARIA language, which provides an abstract SFE language (with a vocabulary independent of the interaction modality) and a set of concrete SFE languages in which the abstract vocabulary is refined taking into account the target interaction platform.

The adaptation process consists of three main steps, implemented by the following modules:

1. The Reverser parses the HTML and CSS implementation, and creates a MARIA model-based, logical description of the SFE, including contents and semantic information such as groupings, heading, navigation etc. (some are already included in the HTML, others are obtained by applying specific heuristics). This description is tied to a graphical desktop vocabulary.

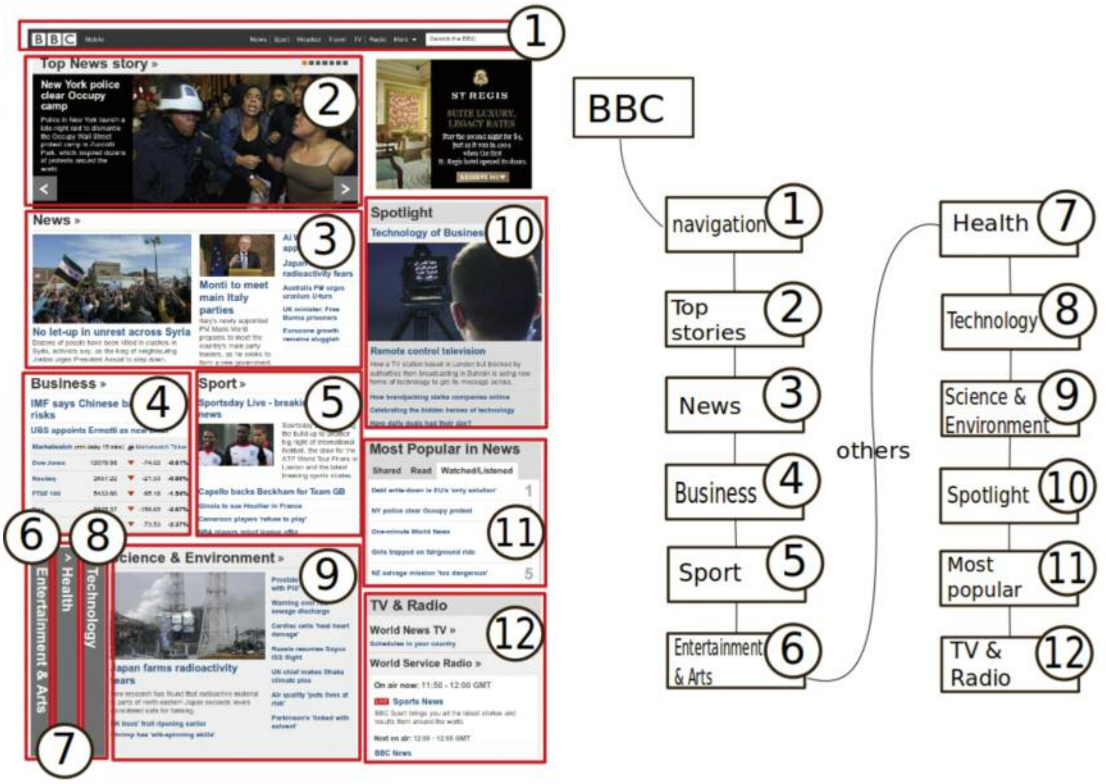

Figure 1: Cross Modality adaptation example

2. The Graphical-to-Vocal Adapter transforms the Reverser output into another MARIA model-based, logical description, switching the vocabulary from graphical desktop to vocal. This step consists of three sub-steps. The first performs a content and structure optimization, removing elements that are only used for layout purposes and splitting the contents so that they can be easily browsed vocally. The second step creates the vocal menus that support the interface navigation, finding meaningful labels for the identified content groups. The last transforms the desktop elements into vocal ones (ie a drop-down list to a vocal choice among a predefined set of options, images to alternative text descriptions, submit buttons to confirmation prompts, etc.).
3. The Voice XML Generator takes as input a MARIA SFE description for the vocal platform and translates it into a Voice XML application that can be accessed through vocal browsers.

Since the adaptation is performed on the logical SFE descriptions it is independent of the implementation languages. In addition, the adaptation process can be customized by SFE designers through a tool that allows a set of parameters to be changed (eg the maximum number of items in a vocal menu). The menu structure that will be generated can also be previewed and updated when a parameter has been changed.

Figure 1 shows an example of the desktop-to-vocal SFE adaptation on the BBC News site. The left part represents the original page layout. The adaptation process identified a set of different groups, which are highlighted with the red boxes. It then created the navigation structure, shown in the right part of the figure. The text within the boxes shows the words the user has to pronounce in order to move to the corresponding section. Finally, each page element is transformed into its vocal counterpart.

The resulting application will start asking the user to choose between the first set of parts in the BBC homepage (Navigation, Top stories, News, Business, Sport, Entertainment & Arts) or others. If, for instance, the user says “others” then the remaining parts of the page will be listed (Health, Technology etc). If the user says “News”, the application will start reading the latest news included on the BBC website. The user is always able to go back to the previous menu, restart the navigation or exit the application by pronouncing the words “Back”, “Previous” or “Exit”, respectively.

We have presented a solution to make Web pages more suitable for vocal browsing by analyzing and modifying their logical structure. A customization tool was also developed in order to interactively change the parameters of the adaptation process and have a pr view of the structure of the generated vocal application.

Links:
HIIS Lab: http://giove.isti.cnr.it
MARIA home page and Environment: http://giove.isti.cnr.it/tools/MARIA/home
SERENOA EU Project: http://www.serenoa-fp7.eu/

Please contact:
Fabio Paternò
ISTI-CNR, Italy
E-mail: fabio.paterno@isti.cnr.it

CAPucine: Context-Aware Service-Oriented Product Line for Mobile Apps

by Carlos Parra, Clément Quinton and Laurence Duchien

The design of a mobile application is a tedious task owing to its intrinsic variability. Software designers must take into account in their development process the versatility of available platforms (eg Android, iPhone, tablets). The variety of existing devices and their divergences (eg frontal camera, GPS) introduce a further layer of complexity to the development process. In addition, at runtime, many potential situations have to be considered (eg limited connectivity, hardware heterogeneity, changes of user preference, etc.). Such software systems are seen increasingly as evolutive systems.

To tackle such a challenge, a cornerstone element that is intrinsic to any modern software is the notion of adaptation. If a system is designed and implemented to be adapted, then it is more likely that it can better support changes in its requirements, architecture, and even implementation. Software adaptations open the door for new categories of systems that use all the information available both at design time, to postpone and minimize the impact of developer decisions, and at runtime, to take advantage of the information available in the application environment. However, adaptations are difficult to define since they may take place at early stages of the development process, but also at runtime where there are many situations that have to be considered. Typically, this kind of information is designated as context information. Hence, a context-aware system is aware of changes in the context and is able to adapt to offer better user experiences. A well-known example of such behaviour is when a system changes its behaviour depending on the location. This is especially true in mobile computing.

In this project, we explore the applicability of the Software Product Line (SPL) paradigm to the development of adaptable systems for mobile applications. SPLs aim at managing and building multiple software products from a set of previously developed and tested assets. An asset is understood as any software artifact that can be