Adaptive Mobile Multimodal User Interfaces for Dynamic Accessibility

Marco Manca, Fabio Paternò, Carmen Santoro
CNR-ISTI, HIIS Laboratory
Via Moruzzi 1
56124 Pisa, Italy
{marco.manca, fabio.paterno, carmen.santoro}@isti.cnr.it

CHI 2013, Paris
Workshop on Mobile Accessibility

Abstract

In this paper we discuss how to exploit solutions able to support mobile multimodal adaptive user interfaces for dynamic accessibility. We focus on an approach based on the use of declarative user interface languages and oriented to Web applications accessed through emerging ubiquitous environments.

Keywords

Adaptive User Interfaces, Model-base user Interface Languages, Mobile Users, Accessibility

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms

Design

Introduction

Emerging ubiquitous environments call for supporting access through a variety of interactive devices. Web applications are widely used and a number of approaches and techniques have been proposed to support their adaptation to mobile devices. Little attention has been paid so far on identifying more general solutions able to adapt Web applications to various combinations of mobile multimodal user interfaces. We have developed a solution for this purpose that aims to support any Web application independent of the type of authoring environment used for its development. We have considered adaptation to various possible contexts of use (varying in terms of interactive devices, user preferences, environmental aspects, ..) and the approach can be useful for supporting disabled users.

Adaptation in mobile contexts is a topic that has recently stimulated various research contributions. For example, W3Touch is a tool that aims to collect user performance data for different device characteristics in order to help identify potential design problems for touch interaction. In mobile devices it has been exploited [5] to support adaptation based on two metrics related to missed links in touch interaction and zooming issues to access Web content. However, such work has not considered accessibility issues.

A work that has similar goals as ours is DANTE [4], which provides semantic transcoding for visually-disabled users. For this purpose it uses ontology-based annotations that are manually created and included in the Web page. Such annotations are then used for automatically transforming the original pages to obtain pages more easily accessible for visual-impaired users. In our case, we want to provide a tool that does not require any manual intervention and is able to automatically generate multimodal Web applications for the target device. For this purpose we consider intermediate model-based representations [3], which aim to represent the logical structure of the user interface and enable relevant reasoning to derive a more suitable structure for the target device. An attempt to exploit model-based languages in this direction was presented in [1], however that work considered the use of ontologies in identifying the choice of the interaction techniques, and this makes the approach not suitable for interactive environments where the dynamic adaptation results should be presented efficiently. Concepts related to model-based approaches have recently been considered in the Indie working group [7] in order to make it easier for Web applications to work in a wide range of contexts through an abstraction between physical, device-specific user interaction events and inferred user intent such as scrolling or changing values. This can provide an intermediate layer between device- and modality-specific user interaction events, and the basic user interface functionality used by Web applications.

In this paper we discuss our approach and how it can be exploited to obtain multimodal adaptation in mobile devices for supporting disabled users.

Architecture of the Solution

We exploit the model-based framework MARIA [6] for performing a more semantically-oriented transformation. The framework provides abstract (independent of the interaction modality) and concrete (dependent of the interaction modality but independent of the implementation languages) languages. Such languages share the same structure but with different levels of refinements. A user interface is composed of a number of presentations. In each presentation there are composition elements (e.g. groupings, repeaters) that identify the main logical parts and contain the interface elements, which are called interactors. Examples of interactors are navigators (allow moving from one presentation to another), activators (allow triggering functionality), …

MARIA is an engineered language able to describe any interactive feature of Web applications, and also access to remote Web services. The MARIA specification of interactive applications can be obtained either through publicly available authoring environments[1] or through reverse engineering tools able to build such specifications starting with a HTML(5) + CSS(3) implementation. In the case of HTML5 the reverse process is facilitated by its use of tags that provide more information on the structure of the page and the role of its elements.

[1] http://giove.isti.cnr.it/tools/MARIAE/home

In addition, we have defined a language to specify adaptation rules in terms of events, conditions and actions. Events indicate things that happen in the context of use (e.g. the noise or the light has increased over a certain threshold) or in the user interfaces (e.g. the user has changed the zoom level). Conditions refer to some specific state in the context or the user interface that should be verified. Actions describe the effects of the adaptation one the user interface, which can be specified either in its logical description, from which then derive the corresponding implementation, or directly in its implementation.

Our solution is based on an adaptation server (see Figure 1), and provides a number of functionalities:

- *Monitor context changes*, it communicates with a context manager, which is able to collect events from the surrounding devices and environment, for this purpose the adaptation engine subscribes to receiving the contextual events relevant for the current adaptation rules;
- *Select adaptation rules,* it selects adaptation rules that are associated with the events that occur in the context of use and (optional) specific conditions*;*
- *Identify changes*, the results of the adaptation rules are changes in the logical specifications of the interactive applications and/or in their implementations, which are transmitted to the application installed in the mobile device;
- *Update implementations*, the elements in the application implementation in the mobile device corresponding to the changes to perform are updated accordingly.

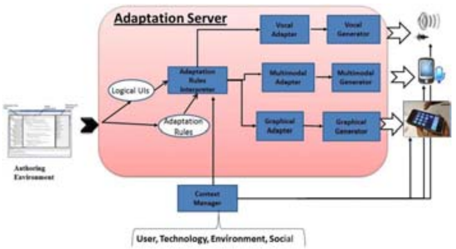

figure 1. The architecture of our solution.

The context manager that we have considered is described in [8], and it has a client-server architecture in which information gathered through sensors can be detected and communicated to the context manager server, which provides it to the adaptation server, in case it has subscribed to the corresponding events. The context manager also contains information regarding user profiles that can be relevant for adaptation. The adaptation server communicates with the applications in the mobile devices that have to adapt according to the adaptation rules triggered.

Support for MultiModal User Interfaces

The MARIA concrete multimodal language uses the CARE properties in order to identify how the various modalities are allocated. We interpret such CARE properties in the following way: *complementarity*, when the considered part of the UI is partly supported by one modality and partly by another one; *assignment*, the considered part of the UI is supported by one assigned modality; *redundancy*, the considered part of the UI is supported by both modalities; *equivalence*, the considered UI part is supported by either one modality or another.

To obtain a flexible definition of multimodal interfaces it is possible to apply such properties to the various UI concepts of the MARIA language: composition operators (groupings, repeaters, …), interaction and only-output elements. In order to have different modalities allocated even at a finer grain, interaction elements themselves can be further structured into prompt, input, feedback, and it is possible to apply different CARE properties to each of such parts.

We interpret the CARE properties from the user viewpoint. Thus, the equivalence property is applied only to the input part of an interaction element when multiple modalities are available in order to indicate the

figure 2. An example application.

possibility to enter the input indifferently through one of these modalities. We do not apply it to output elements because even if the system can choose between two different modalities to present output, in the end only one of them will be provided and perceived by the end user, and so the user will perceive a modality assignment to the output element.

For the generation part, we exploit the Google APIs for vocal interaction that can be accessed also by Web applications. In particular, we have considered Android mobile devices exploiting a Java component (WebView), which is able to show Web pages and access the DOM through JavaScripts. Thus, when a change in terms of multimodality should occur in some part of the user interface, the action part of the adaptation rule indicates what user interface elements should be changed and the new values for their corresponding CARE properties.

Then, there is a set of JavaScripts that are able to identify the DOM elements corresponding to the elements of the logical descriptions indicated in the rule and implement the requested changes by enabling or disabling the graphical and the vocal parts of these elements accordingly to what is specified in the CARE properties.

Overall, the impact of this solution on page browsing in terms of performance (loading time, and consequently the user experience) is limited and allows users to receive information and interact with it through the more appropriate modalities.

Adaptation to Disabled Users

Our approach can address both permanent and temporary disabilities. The case of permanent disabilities is easier to manage since the user profile can immediately indicate the static characteristics of the current user, from which it is possible to derive the most suitable multimodal combination. Thus, for example, vision-impaired users will rely mainly on the vocal modality while deaf users on the graphical one. For temporary disabilities we mean situations in which the user cannot exploit some sense for external circumstances: for example, the surrounding environment becomes noisy and then the acoustic channel cannot be exploited or the user is walking fast and thus the visual channel cannot be exploited too much for the interaction with the mobile application. These dynamic situations can be usually detected by the context manager, which informs the adaptation engine that contains a number of adaptation rules indicating how to change the interaction modalities accordingly.

For example, we can have a rule indicating that when the user starts to walk fast then all the interactive elements should have the input equivalent, entered either graphically or vocally, and the feedback redundant with both modalities. This allows the users to enter the input with their preferred modality and provides them with both graphical and vocal feedback in order to be sure that it can be perceived immediately. In the example in Figure 2 this solution enables the possibility of getting advice through a vocal command without having to touch the *get advice* button, and the advice will then be presented both graphically and vocally.

Conclusions and Future Work

We have presented an approach to Web user interface adaptation that can lead to better accessibility in multi-device environments. We have discussed how it has been designed for supporting mobile disabled users that can interact through various combinations of graphical and vocal modality.

Future work will be dedicated to extending the architectural solution proposed in order to engineering its support for various types of mobile interactive devices, and to carrying out a number of users tests in order to identify a set of adaptation rules for various types of permanent and temporarily disabilities that are empirically validated.

Acknowledgements

We thank the EU SERENOA Project for partly support this work.

References

[1] Abascal, J., Aizpurua, A., Cearreta, I., Gamecho, B., Garay-Vitoria, N., Miñón, R.: Automatically generating tailored accessible user interfaces for ubiquitous services. ASSETS 2011: 187-194

[2] Bongartz S., Jin Y., Paternò F., Rett J., Santoro C., Spano L. D., Adaptive User Interfaces for Smart Environments with the Support of Model-based Language Sara AMI'12 International Joint Conference on Ambient Intelligence, November 2012, Pisa, Lecture Notes in Computer Science 7683 pp.33-48, Springer.

[3] Fonseca J.M.C. (ed.), W3C Model-Based UI XG Final Report, May 2010, available at http://www.w3.org/2005/Incubator/model-based-ui/XGR-mbui-20100504/

[4] Yesilada, Y., Stevens, R., Harper, S. and Goble, C. Evaluating DANTE: Semantic transcoding for visually disabled users. ACM Transactions on Human-Computer Interaction 14, 3, (2007), Article 14.

[5] Nebeling M., Speicher M. and Norrie M. C., W3Touch: Metrics-based Web Content Adaptation for Touch, to appear in Proceedings CHI'13, Paris.

[6] Paternò F., Santoro C., Spano L.D.: MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. ACM Trans. Comput.-Hum. Interact. 16(4): (2009).

[7] Craig J., Cooper M., Events for User Interface Independence, http://www.w3.org/TR/indie-ui-events/

[8] Ghiani G., Paternò F., Santoro C., Context of Use Runtime Infrastructure, SERENOA Deliverable D.4.4.1, http://www.serenoa-fp7.eu/wp-content/uploads/2012/07/SERENOA_D4.4.1.pdf