Multimodality and Multi-device Interfaces

Fabio Paternò

ISTI-CNR
Via G.Moruzzi 1, 56124 Pisa, Italy
fabio.paterno@isti.cnr.it
http://giove.isti.cnr.it/~fabio/

Abstract

This position paper discusses how to take into account multimodality when designing multi-device user interfaces. In particular, it focuses on graphical and vocal interaction and discusses how to obtain interfaces in either one modality or their combination starting with logical descriptions, such as the indications of the tasks to perform in order to reach the user's goals. It also introduces how such an approach can enable achieving context-dependent or migratory interfaces exploiting various modalities.

Introduction

Nowadays, everyday life is becoming a multi-platform environment where people are surrounded by different types of devices through which they can connect to networks in different ways. One of the main current challenges for designers and developers of interactive systems is how to address applications that can be accessed through a variety of devices that may vary in terms of interaction resources (screen size, processing power, modalities supported, …). Multimodality adds further complexity to this challenge: depending on the software and the browser available, the same device can support different modalities in different manners.

To address such issues a key aspect is to be able to have different views of interactive systems, each view associated with a different abstraction level [5]. With the support of tools, XML-based languages and transformations, it is possible to move from one level to another and convert a description for one interaction platform to another for a different one. XForms [6] is an example of W3C standard, which applies some concepts that have been investigated in the research field of model-based design of user interfaces. The potential logical descriptions that are usually considered are: task models representing the logical activities to perform in order to reach users' goals; object models describing the objects that should be manipulated during task performance; abstract user interfaces providing a modality independent description of the user interface in terms of presentations and logical interactors; concrete user interfaces providing a platform-dependent description identifying the concrete interaction techniques adopted. The user interface implements all of the foregoing.

Design issues

High level descriptions can be obtained through languages that are independent of the platform, and consequently of the modality because by platform we mean a class of systems that share the same characteristics in terms of interaction resources. Examples of platforms are the graphical desktop, multimodal PDAs, mobile phones and vocal systems. Their range varies from small devices such as interactive watches to very large flat displays. However, designers should be aware that the choice of the platform can have an impact from the highest level descriptions. There are tasks that are meaningful only on specific platforms. For example, watching a film or a football match does not make sense on a small mobile device, whereas getting real-time road directions while driving can be done only through mobile devices. The task performances can be influenced by the modality

available: a set of input can require separate interactions through a graphical device, whereas such information can be provided through a single interaction by using a vocal interface.
Multimodality is important in this approach because of the intrinsic differences between modalities. For example, the vocal channel is more suitable for simple or short messages, for signalling events, immediate actions, to avoid visual overloading and when users are on the move, whereas the visual channel is more useful for complex or long messages, for identifying spatial relations, when multiple actions have to be performed, in noisy environments or with stationary users.

Authoring environments for Multimodal and Multidevice Interfaces

Since interest in developing multimodal Web interfaces is bound to increase, it is important to provide designers with authoring environments for this purpose. Such authoring environments should allow designers to focus on conceptual aspects and then they should provide intelligent support in order to obtain usable interfaces corresponding to the tasks to support. Unfortunately, most current authoring environments just allow the development of user interfaces from the basic elements. They may provide a number of direct manipulation facilities that allow to easily create interfaces, but they usually do not provide any support to identify the most suitable interface for supporting the target users and tasks, while also taking also into account the potential interaction platforms and modalities. We have developed a method and a tool supporting it (TERESA [4]) that exploits a number of transformations that allow designers to move through various views of interactive systems. TERESA allows designers to focus on the logical tasks to accomplish and then transform their descriptions into a user interface description, which incorporates the design decisions but is still in a modality independent format. This is then used to derive a modality-dependent interface description, which forms the base for generating the final code of the user interface. For each logical level considered an XML-based language has been defined. While the languages for the task and the abstract interface descriptions are the same for all the platforms, each platform has its own description language for the concrete interface. The advantage of this approach is that designers can focus on logical aspects and make design decisions without having to deal with many low-level implementation details. This type of support is particularly useful when a variety of devices should support access to the interactive application. On the other hand, it is worth noting that in order to make effective decisions designers should be aware of the target platforms and modalities right from the early stages of the design process. They do not need to know the implementation details of all the targeted devices, but they still should know their main features (through the platform concept).

While the task model describes the logical activities necessary to reach users' goals, the abstract user interface is structured into presentations. For each presentation, the tool identifies the associated logical interactors and provides declarative indications of how such interactors should be composed. This is obtained through composition operators that have been defined taking into account the type of communication effects that designers aim to achieve when they create a presentation: Grouping, indicates a set of interface elements logically connected to each other; Relation, highlights a one-to-many relation among some elements, one element has some effects on a set of elements; Ordering indicates some kind of ordering among a set of elements; Hierarchy, highlights different levels of importance that can be defined among a set of elements. In addition, navigation through the presentations is defined taking into account the temporal relations specified among tasks. The abstract user interface description can then be refined into a concrete user interface description, whereby a specific implementation technique and a set of attributes are identified for each interactor and composition operator, after which the user interface implementation can be generated. Currently, the tool supports implementations in XHTML, XHTML mobile device, and VoiceXML [2] and a version for multimodal user interfaces in X+V is under development (also EMMA can be considered in the future).

When the goal is to obtain interfaces supporting multiple modalities (for example graphical and vocal interaction), the choice of the implementation techniques still has to consider other aspects of the platform. So, it will be different generating user interfaces for a multi-modal desktop system or for a multi-modal PDA system because in one case the user is stationary and the graphic screen large, whereas in the other case the user can be on the move and the screen is small. Thus, the ways to provide input information or output or feedback on the input have to take into account the features of the available platform. The analysis of the space of the possible design choices and the identification of the most suitable one can be done through the CARE properties [3] that identify various ways to use multimodalities: complementarity (synergistic use), assignment (one specific modality is selected), redundancy (multiple modalities for the same purpose), and equivalence (there is a choice of one modality from a set of available ones).

Context-dependent and Migratory Interfaces

This type of approach can be exploited to obtain usable context-dependent or migratory interfaces. In context-dependent interfaces we have interactive systems that depending on the current context automatically change the set of resources used for interaction. For example, if the system detects that the environment is noisy, then the vocal channel is not used. In migratory interfaces the basic idea is that the interface, to some extent, follows the users moving from one device to another, adapting to the features of the new device without having to restart the session from scratch. This enables multimodal migrating services that, for example, allow the user to start the registration to the service through a desktop system. Then, in the event he realizes that it is getting late, he can carry on the registration on the move using a PDA until he reaches his car where he can use a vocal interface to complete it. All this activity can be performed without having to restart the registration procedure from the beginning at each device change.

This can be achieved through the approach discussed, which allows designers to obtain different versions of application interfaces tailored for each platform. Such versions should be managed by a migration server. The goal of the migration server is to maintain information of the devices available for the migration service and their characteristics, receive requests for migration, identify the target device of the migration and activate a new instance of the interface for the identified device. Such interface should already have the state of the results of the user interactions with the source device. This implies the ability to store the state of the user interactions, convert such state into the state for the target device interface, and identify the specific presentation that should be activated in the target device in order to allow continuity of interaction.

This type of processing can be performed because of the logical information that is created during the model-based design process. The tasks supported by each interface platform can vary or, the way they are supported can substantially change. The migration server knows the associations between tasks and interactors and between interactors and interface elements for each platform. Thus, for each user interaction it can identify the corresponding task and then determine how it is supported by the target device and associate the state generated by the results of the user interactions in the source device to the corresponding interface elements in the target device. In addition, the knowledge of the last task performed on the source device is useful information also for identifying the presentation in the target device that should be activated as a result of the migration because the user expects to carry on his session from that point. This type of approach can be applied to migratory interfaces that change modalities when moving from one device to another [1].

Conclusions

In this position paper I have discussed the design issues raised by multimodal interface in the context of ubiquitous computing. I have indicated a number of methods and tools based on the use of logical descriptions for obtaining suitable authoring environments for multimodal Web interfaces and discussed how this approach can also be useful to obtain context-dependent or migratory interfaces.

Future work will be addressed to considering a broader set of modalities in this approach and also to extending the authoring environment proposed by exploiting multimodality. Thus, it is hoped that designers will be able to work with more familiar and understandable representations in more natural development environments

Acknowledgments

This work has been partly supported by the SIMILAR European Network of Excellence on Multimodal Interfaces of the IST Program of the European Union (www.similar.cc)

References

1. R.Bandelloni, S.Berti, F.Paternò, Mixed Initiative, Trans-Modal Interface Migration, Proceedings Mobile HCI 2004, Springer Verlag LNCS.
2. S. Berti, F. Paternò, Model-based Design of Speech Interfaces, Proceedings DSV-IS 2003, Springer Verlag, LNCS 2844, pp.231-244.
3. J.Coutaz, L.Nigay, D.Salber, A.Blandford, J.May, R.Young, Four Easy Pieces for Assessing the Usability of Multmodal Interaction: the CARE properties. Proceedings INTERACT 2005, pp.115-120
4. G. Mori, F. Paternò, C. Santoro, Design and Development of Multi-Device User Interfaces through Multiple Logical Descriptions, accepted for publication on IEEE Transactions on Software Engineering, 2004.
5. F.Paternò, Models for Universal Usability, Proceedings IHM’03, ACM Press, pp.9-16, Caen, November 2003.
6. XForms – The Next Generation of Web Forms, http://www.w3.org/MarkUp/Forms/