Ambient Intelligence and Multimodality

Laura Burzagli, Pier Luigi Emiliani, and Francesco Gabbanini

IFAC-CNR, Via Madonna del Piano, 10, Sesto Fiorentino, Firenze, Italy
{l.burzagli, p.l.emiliani, f.gabbanini}@ifac.cnr.it

Abstract. Ambient Intelligence (AmI) scenarios place strong emphasis on the fact that interaction takes place through natural interfaces, in such a way that people can perceive the presence of smart objects only when needed. As a possible solution to achieving relaxed and enjoyable interaction with the intelligent environments depicted by AmI, the ambient could be equipped with suitably designed multimodal interfaces bringing up the opportunity to communicate using multiple natural interaction modes. This paper discusses challenges to be faced when trying to design multimodal interfaces that allow for natural interaction with systems, with special attention to speech-based interfaces. It describes an application that was built to serve as a test bed and to conduct evaluation sessions in order to ascertain the impact of multimodal natural interfaces on users and to assess their usability and accessibility.

1 Introduction

According to the Ambient Intelligence (AmI) paradigm, present-day computers and telecommunication terminals will be replaced by an environment in which people will be surrounded by intelligent and intuitive interfaces. The environment will be aware of human needs and will be capable of proactively helping people to organize, structure, and manage their everyday activities in an invisible and unobtrusive way (see [5]).

AmI is supposed to support humans to reach their goals by engaging them in "natural" dialogues, possibly through the introduction into the intelligent environment of suitably-designed multimodal interfaces that offer the opportunity for communicating using multiple natural interaction modes such as speech, eye gaze, gestures and haptics, thus simulating the way in which human beings communicate with each other. While it will take some time to see full-featured AmI scenarios in place, the time seems mature for exploiting the capabilities of multimodal interfaces with a view to their future integration in AmI environments. In addition, interest in multimodality has recently been renewed due to the diffusion of mobile devices and to the awareness that multimodal interfaces can make, right from the present day, a valuable contribution to achieving universal access in the information society. Thus, the importance of multimodality is twofold: in the short term, it will make its contribution to providing universal access in the information society as it is structured at present; in the long term, it will serve as an essential foundation for AmI environments. However, benefits will not come automatically with the introduction of multimodal interfaces: they

C. Stephanidis (Ed.): Universal Access in HCI, Part II, HCII 2007, LNCS 4555, pp. 33–42, 2007.

will show up only if users and modalities are central during the design process. This paper attempts to outline in section 2 what challenges are to be faced when trying to design multimodal interfaces that allow for natural interaction with systems, with a closer look to speech based interfaces.

Section 3 describes an application that is based on an architecture which was introduced in [1]: it was built to serve as a test bed and to conduct evaluation sessions in order to ascertain the impact of multimodal natural interfaces on users and to assess their usability and accessibility.

2 Multimodal Interfaces

Multimodal interfaces are considered useful for interaction with AmI, especially when e-inclusion problems are taken into account. In particular, when interacting with services traditionally based on a direct human interrelation, a "human-like" dialogue could be a suitable solution to cooperate and get support from AmI, avoiding the need of learning and adapting to machine oriented user interfaces. This could be the case, for example, of the prototype service described in the paper (see section 3), which is meant to test features of multimodal interaction.

2.1 General Perspectives and Challenges

Interest in multimodality has recently been renewed[1] thanks also to the diffusion of mobile devices (which, in an evolutionary perspective, could evolve to become examples of future smart objects in AmI). Due to the fact that, at present, graphical interfaces and limited screen size oblige users to follow complex approaches in order to accomplish simple tasks, users are under-using mobile applications (as reported in [6], for example). Providing applications with multimodal interfaces might become a way to achieve more efficient, pleasant and natural interaction (see [10]). Moreover, there is general awareness that the flexibility and interaction capabilities offered by multimodal interfaces can make a valuable contribution to achieving universal access in the information society (see [9, 2, 11, 8]). All systems are supposed to achieve better usability and accessibility if users are able to interact with them through multimodal interfaces; therefore, multimodality may be of key importance for e-inclusion, by improving the usability and accessibility of information systems for a vast diversity of users, including elderly people. While there is general agreement on the fact that multimodal interfaces can make a significant contribution to rendering interaction easier, that they are easier to learn, and are preferred by users for many applications, it is worth considering that these statements are not automatically and inherently tied to injecting multimodality into interfaces. They will show up if the user and the modalities are kept central during the design process. So far, the development of user interfaces has been a technology-driven, rather than a user-driven, process: software design expects users to go through a learning and training process, to finally adapt to the product capabilities. This is especially true as computing capabilities are now

[1] In Europe the Similar network of excellence (http://www.similar.cc) comprises groups with interest in multimodality.

migrating away from the ordinary desktop computer towards mobile objects such as cell phones or PDAs. Due to limited screen size, interfaces tend to be rather complicated and it is almost inevitable that much time is spent by users in applying their attention to gaining a certain understanding of how mobile applications work. Instead, according to the modern approaches to interaction design, users should not have to adapt to machines, and machines should support users to achieve their goals. As for multimodal interfaces, this assumption has a number of implications. First of all, interfaces have to be designed from a user-centered perspective, so as to maximize user performances (along with system performance) and to satisfy user preferences. In this respect, Oviatt observes in [10] that multimodal interfaces are built using recognition-based technologies which have to interpret human behaviors that often are not even under full conscious control of users. Thus, it is virtually impossible for users (even for the most cooperative of them) to adapt to this kind of interfaces. Secondly, to achieve better matching of modalities to user needs, the context must be well defined and properly modeled. New approaches towards context modeling (see [3, 4]), could allow better shaping human activities, more efficiently anticipating their needs, more appropriately capturing attitudes to interact using different modalities and more accurate semantic interpretation of multimodal utterances. A third important point is that, in multimodal applications, the interface must be designed by respecting each modality's peculiarities, expressiveness, and interaction capabilities. Thus, it is necessary to understand how to reorganize and transform information in order for it to be appropriately conveyed by the desired modality, paying attention that the information contents are preserved across different modalities, at the maximum extent possible. This is a very difficult task, and it requires a profound understanding of the capabilities of each modality, which still need to be analyzed in depth. Actually, different modes have their own peculiarities, and these are best suited to conveying different types of information. No complete analogy exists between the different modes. For example, a speech interface is not the best one for describing the contents of a geographical map in a GIS application; it is, however, probably the most efficient if it is used in a conversational application such as a virtual travel agency or a booking centre for making appointments for medical examinations. This third aspect, along with the first one mentioned, implies that, when designing multimodal interaction, human computer interaction researchers should work as a team together with experts in different research fields, including the cognitive sciences. Moreover, it is important to keep in mind that no particular modality should be treated as a reference modality. This is what happens now in many voice-enabled applications, in which the graphical user interface has the role of "main interface" and the speech interface is treated as secondary, and it translates and describes information conveyed by windows and forms, thus loosing expressive power and giving rise to applications with poor usability. If these issues are not considered, few benefits can be expected from the introduction of multimodal interaction in applications, even if, from the implementation point of view, for some modalities, the technology seems mature enough to provide designers with tools that have the potential to bring a satisfactory and efficient multimodal interaction. In section 2.2, the principles that were discussed here at general level, in a modality independent fashion, are detailed in relation to a particular modality. The

speech modality is chosen, because it is one of the most familiar to humans and because knowledge about speech interaction, both from the technological and the human factors points of view, appears to be sufficient to implement system that aim at achieving natural and human-like interaction.

2.2 Speech Based Interfaces

Among all the communication modalities, speech is probably the one to which humans are best adapted: people use their voices to exchange ideas and to communicate in their everyday life. Thus, human-computer speech interfaces have a great interaction potential, right from the present day, considering the fact that (apart from GUIs) they are the ones for which technology is more mature and readily available.

Though, enjoyable speech interfaces can be obtained if designers do not incur in the error to create them based on GUI metaphors and treat the GUI as the reference modality: there is general agreement, coming from past studies (see [7, 15] for example) that translating GUI results in speech interfaces with poor flexibility and usability. Speech interfaces can be roughly divided into two groups: menu driven and conversational ([7]).

Menu driven systems currently represent the most common type of applications with speech interaction capabilities. They enable users to accomplish their tasks by navigating through a hierarchy of successive steps. At each step, interaction possibilities are defined by a series of choices coded in menus, and users express their choice by issuing a vocal command. This theoretically represents an advance over keypad interfaces, where choices are expressed by pressing a number key, because menu driven voice systems allow a richer vocabulary and are more semantically consistent in that they eliminate the arbitrary association between a key and its corresponding function. However, the drawbacks are that their structure is so inflexible that they become tedious to use, and, moreover, their usability is poor when there are too many menu choices, because users' short-term memory gets overloaded. Even if menu driven voice systems are currently used with success in many services offered to the general public (for example they are used in many call centres), they do not appear to put users at the centre of the interaction process; moreover, going through a long series of menu options is indeed a kind of interaction that poorly respects the peculiarities of the speech modality. Given that humans interact with devices to achieve a goal (see [4]), which is certainly not represented by interacting with the computer system itself, the menu driven approach implies modelling tasks necessary to reach the goal state as a precise sequence of user actions, which is common and appropriate in GUI design. Now, it is to be observed that this is not the way humans act in the real world and that actions to be performed when trying to achieve some goal could be better modelled as an unordered set. Spoken dialogues between humans reflect the fact that humans do not usually follow a linear approach to reach their goals and tend to accomplish this by a continuous cooperation with the person who takes part in the conversation. For these reasons, conversational systems appear to offer a better model of users' behaviour and of the modality's characteristics.

During interaction with these systems, rather than asking a series of questions, the computer is supposed to cooperate with users (and not the inverse) in achieving some result. Interaction is more natural, because users can condense a sequence of menu

choices into a single language utterance, which obtains its meaning both from the context and from the spoken words. This puts the user at the centre of the interaction process and highlights the importance of an appropriate modelling of context. In fact, certain expressions might have different meanings, depending on different situations. These reasons make conversational interfaces suitable for use in AmI environments, to enable access to dialogue based services. Though, designing conversational interfaces is a hard task, because it requires a knowledge in fields related to technology and computer science, as well as in fields that regard human-factors, such as linguistics, philosophy, psychology, sociology, in order to properly understand the mechanisms underlying human-human dialogues and to catch their essence. From a technological perspective, the main problem arises from the fact that dialogue inputs are uncertain, since speech recognizers have their error rates, so that one of the biggest challenges is to design interfaces that maximize the probabilities of correct understanding. To achieve this, great help comes from understanding the vocabulary, and this is attained from careful study and knowledge of the language domain in which a speech-based system will operate. Clearly, ordering a pizza is different from buying opera tickets: each domain has its own terminology, utterance structures, and interactive patterns. Also, it is important to build a model of the discourses that users employ in communicating and in performing their tasks. These can be key factors in obtaining usable, enjoyable and robust conversational interfaces.

3 System Overview

A speech based component in a multimodal interface is considered important for interaction with AmI, especially, but not only, when accessibility problems are taken into account. This is particularly true when services traditionally based on dialogues with humans need to be accessed, and this is the main reason why a prototype of a service based on speech interaction was regarded as an interesting building block of multimodal interfaces, to test some of the general considerations about the proper use of the features of the different modes.

3.1 Architectural Requirements

Designing and implementing “natural” multimodal systems is a difficult task, from both the technological and the human-factors perspective. It requires the use of suitable architectures. Their key factors should be modularity and, as pointed out in [12], the capability of easily integrating new technologies, in order to grow smoothly and facilitate evolutionary progress.

It is indeed necessary that developer teams include human computer interaction experts, as well as experts from all the fields pertaining to the modalities that the interface has to comprise (such as lexicographers and psychologists with discourse analytic training in the case of speech interfaces). If the architecture is well specified at an abstract level, the development effort can be distributed among different teams, each one focusing attention on designing the interaction for different modalities; contributions can then be integrated to form the final multimodal interface for the system.

As for architectures, a generic design approach for the development of multimodal applications that appears convincing is the one introduced by Carbonell and described in detail in [1]. It is based on a six-layer software architecture that aims to facilitate multimodal input and output processing, by matching multimodal utterances of users to appropriate methods supported by the application in question. One of the strengths of this approach lies in the fact that it neatly separates the user interface layer from the rest of the system, thus providing the possibility of designing unimodal interfaces separately from the others, possibly using, at the time of implementation, software for the generation of monomodal output and interpretation of monomodal input available on the market, and of developing the system in a cooperative fashion.

3.2 Application Description and Implementation Issues

The test application regards a public utility scenario which is potentially useful for the vast majority of citizens. It allows interaction with a booking and management service for medical examinations through a multimodal interface that combines a classical GUI with a vocal interface capable of speech recognition and synthesis. It is possible to check for available dates for medical examinations, to make, review and delete appointments, and to inspect results for examinations that were already taken. The application is based on an architecture that is essentially the one specified in [1] and described in section 3.1. It was built mainly to evaluate multimodal natural interfaces, to assess their usability and accessibility, and to evaluate their potential uses in AmI environments. While the focus of this paper is not on technology, technological aspects have naturally been considered to be important and have been exploited during the development process: in this respect, attempts have been made to design a modular application that allows for the reuse of important architectural components, such as monomodal output generators and monomodal input interpreters, with a view to future work.

During implementation, it was considered important to have the possibility for easily plugging modalities in and out, in order to better investigate how people make use of different combinations of modes, and, as previously mentioned, to facilitate collaborative work so that modalities can be implemented by different teams.

Some relevant implementation details about the application, whose overall structure is represented in Fig. 1, are reported in the following of this section.

As for the platform used, it is to be considered that the first interfaces to be integrated in the application were a GUI and a voice interface. Software development kits that provide programmers with a rich set of tools for building speech dialogue interfaces are available on the market: some examples are the Java speech API[2], the Microsoft .NET Speech Technology[3], the Philips Speech SDK[4], the IBM Via Voice SDK[5]. These toolkits all include accurate and robust speech recognizers that are reliable enough to meet general usability requirements for a speech-based interface. To achieve quicker integration, it was decided to use the Microsoft .NET 2.0 platform and Microsoft .NET Speech Technology. The platform provides assemblies that, besides

[2] http://java.sun.com/products/java-media/speech/
[3] http://msdn2.microsoft.com/en-us/netframework/default.aspx
[4] http://www.speechrecognition.philips.com
[5] http://www-306.ibm.com/software/voice/viavoice/dev/

integrating APIs that facilitate the development process of graphical user interfaces, offer the opportunity for easily integrating speech synthesis and recognition in an application. This makes it possible to focus attention on the interaction design, rather than on other implementation details, while providing an opportunity for later coding or integration of other monomodal interpreters/generators (either packed in assemblies compatible with the .NET platform or in DLLs compiled from C/C++ code). To address the general requirement expressed in [1] that a generic multimodal interface should offer the possibility for easily plugging in/out additional/existing modalities, different modalities were implemented in distinct assemblies.

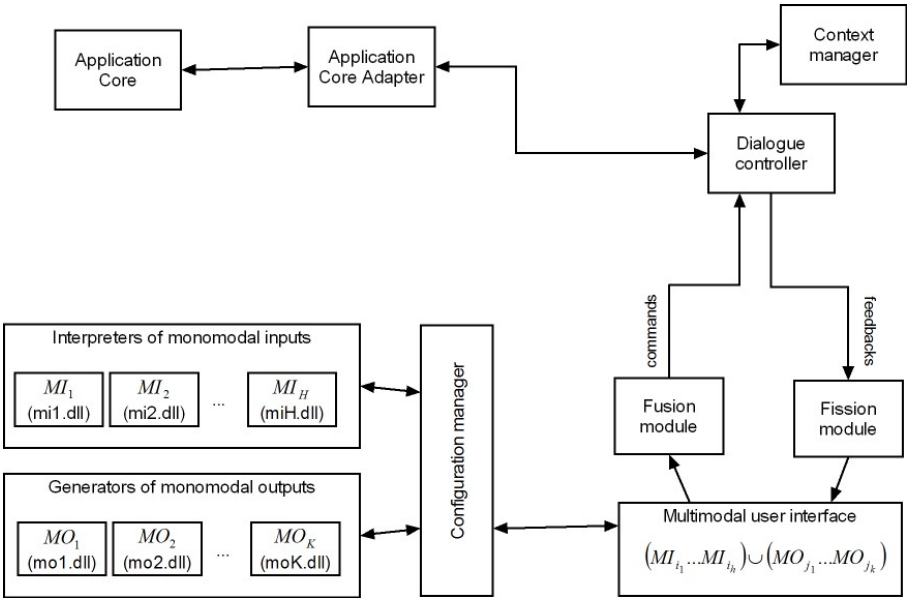

Fig. 1. Application structure

The decision as to which modalities should be included in the user interface is at the moment taken at start up time, during a configuration process based on an XML file that specifies which modalities to use, where the corresponding assemblies are to be found and how to configure the modalities. Anyway, nothing prevents loading and unloading interfaces while interaction occurs, for example as a result of adaptation processes. The configuration manager then employs dynamic loading and late binding, using reflection services (which, in the .NET universe, denote the process of runtime type discovery) to start up modalities of which it has no compile time knowledge of. While the GUI development did not require major efforts, when designing the speech interface the main point on which attention was focused was that of enabling conversational style interaction. To achieve reliable natural interaction with a speech system, it is first of all important to limit the speech recognizer's input uncertainty.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<grammar root="check_dates"
         xmlns="http://www.w3.org/2001/06/grammar"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-
         instance"
         xsi:schemaLocation="http://www.w3.org/2001/06/
         grammar http://www.w3.org/TR/speech-      gram
         mar/grammar.xsd" xml:lang="en-US" ver
         sion="1.0">
         <rule id="check_dates">
           <example>I would like to check for available
         dates for a medical examination </example>
           <example>I would like to make an appointment
         for a medical examination </example>
           <item repeat="0-1">please</item>
           <item>
             <item repeat="0-1">I would like to</item>
             <one-of>
               <item>
                 <item> check for </item>
                 <item repeat="0-1"> available </item>
                 <item repeat="0-1"> dates for </item>
               </item>
               <item> make an appointment for </item>
             </one-of>
             <one-of>
               <item> an examination </item>
               <item> a medical examination </item>
             </one-of>
           </item>
           <item repeat="0-1"> please </item>
           <tag>$.out = "check_examination";</tag>
         </rule>
</grammar>
```

Fig. 2. Example of grammar that allows phrases such as: “I would like to check for available dates for a medical examination” or “I would like to make an appointment for a medical examination”. The semantic content is specified through the tag declaration.

It is then to be recalled that people are accustomed to speaking to other people, and the design must account for this preconditioned input method and take it as an “inspiration” to enable users to engage in natural and intuitive conversations with the system. As the platform supports W3C Speech Recognition Grammar Specification[6] (SRGS), it was decided to rely on grammar in order to achieve a balance between reliability and the possibility for users to pronounce articulated word sequences (see Fig. 2 for an example). Grammars basically define what the system is able to hear and interpret from the user at each segment of a dialogue, and is intended for use by speech recognizers and other grammar processors so that developers can specify the

[6] http://www.w3.org/TR/speech-grammar/

words and patterns of words to be heard by a speech recognizer, thus mitigating the effect of input uncertainty. In order to define grammars, it is necessary to properly understand the vocabulary (as described in section 2.2) by obtaining knowledge of the language domain. Even if it is typically outside the scope of the speech recognizer (and not supported by SRGS), but within the scope of a dialog manager (with the help of information coming from the context), to perform a more thorough semantic analysis, the SRGS provides syntactical support for a form of semantic interpretation through the use of the `tag-format` and `tag` declarations.

The use of grammars can help making interaction as natural and conversational as possible while reducing error rates during the recognition process. They are important to support specific groups of users with different speaking styles, such as accented people, children or elderly people, in which recognition error rates are higher (see [14]). This assumption is supported by the fact that, during our preliminary test sessions, the speech recognizer did not have any problems with users speaking English with a strong Italian accent.

4 Conclusions and Future Directions

The paper describes challenges to be faced and outlines possible directions to be followed when designing applications with multimodal interfaces that allow for natural interaction with computers, in view of future integration in AmI. In section 3, it discusses an example application designed according to the outlined directions. Of course, the application does not address all the themes involved in AmI: it deals with aspects related with natural multimodal interaction that were considered important for the development of AmI. As the application is now in place and ready to be tested, interest is in programming evaluation sessions to ascertain the impact of multimodal interaction on users, with special interest on visually impaired users, in order to check usability and accessibility. Also, it will be interesting to analyze and introduce more modalities into the application.

With a view to AmI, future directions of research will investigate the introduction of multimodal capabilities into portable devices, to allow for interaction with forms of intelligence distributed in the environment, for example by establishing wireless connections of PDAs to desktop computers playing the role of AmI. In this direction, it could be interesting to study possible benefits coming from the addition into the framework of a module handling abstract descriptions of the user interface like the ones described in [13].

References

1. Carbonell, N.: Multimodal interfaces -A generic design approach. In: Stephanidis, C. (ed) Universal Access in Health Telematics. LNCS, vol. 3041, pp. 209–223. Springer, Heidelberg (2005)
2. Carbonell, N.: Ambient multimodality: towards advancing computer accessibility and assisted living. Universal Access in the Information Society 5(1), 96–104 (2006)
3. Coutaz, J., Crowley, J.L., Dobson, S., Garlan, D.: Context is key. Commun. ACM 48(3), 49–53 (2005)

4. Crowley, J.L., Coutaz, J., Rey, G., Reignier, P.: Perceptual components for context aware computing. In: UbiComp '02: Proceedings of the 4th international conference on Ubiquitous Computing, London, UK, pp. 117–134. Springer, Heidelberg (2002)
5. Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J.C.: Scenarios for ambient intelligence in 2010. Technical report, Information Society Technologies Programme of the European Union Commission (IST) ((February 2001)
6. Fan, Y., Saliba, A., Kendall, E.A., Newmarch, J.: Speech interface: An enhancer to the acceptance of m-commerce applications. In: ICMB '05: Proceedings of the International Conference on Mobile Business (ICMB'05, pp. 445–451. IEEE Computer Society Press, Los Alamitos (2005)
7. Harris, R.A.: Voice Interaction Design: Crafting the New Conversational Speech Systems. Morgan Kaufmann, San Francisco (2005)
8. Maybury, M.T.: Universal multimedia information access. Universal Access in the Information Society 2(2), 96–104 (2003)
9. Oviatt, S.: Multimodal interfaces. In: Jacko, J., Sears, A., (eds.) Handbook of Human-Computer Interaction. Lawrence Erlbaum, New Jersey, pp. 286–304 (2003)
10. Oviatt, S.: User-centered modeling and evaluation of multimodal interfaces. In: Proceedings of the IEEE vol. 91(9) pp. 1457– 1468 (2003)
11. Richter, K., Hellenschmidt, M.: Position paper: Interacting with the ambience: Multimodal interaction and ambient intelligence. In: W3C Workshop on Multi-modal Interaction, 19./20.07.2004, Sophia Antipolis, France (2004)
12. Russell, D.M., Streitz, N.A., Winograd, T.: Building disappearing computers. Communications of the ACM 48(3), 42–48 (2005)
13. Trewin, S., Zimmermann, G., Vanderheiden, G.: Abstract representations as a basis for usable user interfaces. Interacting with Computers 16, 477–506 (2004)
14. Wilpon, J.G., Jacobsen, C.N.: A study of speech recognition for children and the elderly. In: Acoustics, Speech, and Signal Processing, ICASSP-96. Conference Proceedings. vol. 1 (7-10 May 1996) pp. 349–352 (1996)
15. Yankelovich, N., Levow, G.A., Marx, M.: Designing SpeechActs: issues in speech user interfaces. In: CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co, pp. 369–376 (1995)