Design and Development of Speech Interaction: A Methodology

Nuno Almeida[1,2], Samuel Silva[1], and António Teixeira[1,2]

[1] Institute of Electronics and Telematics Engineering, University of Aveiro, Portugal
[2] Dep. of Electronics, Telecommunications and Informatics Engineering, University of Aveiro, Portugal

Abstract. Using speech in computer interaction is advantageous in many situation and more natural for the user. However, development of speech enabled applications presents, in general, a big challenge when designing the application, regarding the implementation of speech modalities and what the speech recognizer will understand.

In this paper we present the context of our work, describe the major challenges involved in using speech modalities, summarize our approach to speech interaction design and share experiences regarding our applications, their architecture and gathered insights.

In our approach we use a multimodal framework, responsible for the communication between modalities, and a generic speech modality allowing developers to quickly implement new speech enabled applications.

As part of our methodology, in order to inform development, we consider two different applications, one targeting smartphones and the other tablets or home computers. These adopt a multimodal architecture and provide different scenarios for testing the proposed speech modality.

Keywords: Speech, multimodal architecture, decoupled modalities.

1 Introduction

Speech is, in many situations, the easiest and most natural existing interface to deal with computers, not only for people with special needs, but for people in general [18]. The advantages of speech, as argued by Bernsen [6], are many: a) it is natural and so, people communicate as they normally do; b) it is fast (commonly 150–250 word per minute); c) it requires no visual attention; and d) it does not require the use of hands. Adding to these, one of the characteristics that distinguishes the auditory from the visual channel is its omni directionality, i.e., auditory information can be received from any direction and can also, to some extent, be transmitted in parallel with stimuli from other channels. Furthermore, auditory information, even though it is transient, has a slightly longer short-term storage than visual information which allows delayed processing [20].

Using speech for interaction requires the consideration of different components including speech recognition, text-to-speech, grammar management, a natural language generator and adaptability management, possibly considering multiple

M. Kurosu (Ed.): Human-Computer Interaction, Part II, HCII 2014, LNCS 8511, pp. 370–381, 2014.

languages. Some components are inter-dependent and must communicate between them and with the application. One major challenge is to have a flexible design to enable communication and to support a loosely coupled and distributed architecture, allowing an easy integration with application and devices.

Furthermore, one of the most challenging aspects of speech interaction is dealing with users' expectations, as they often expect speech enabled systems to be capable of understanding much more commands than they actually do.

Using speech as an input/output modality should not be done lightly and the literature provides several guidelines [19,15]) that should be considered, covering when to use speech, what kind of tasks and data are best served by speech, how to combine speech with other modalities and how to address adaptability (e.g., to context). One important aspect to note, for example, is that speech should not be used alone, but as part of a multimodal approach, even though, sometimes, it might be the only useful modality for some users or contexts [21]. This integration with other modalities is also a challenging task [9].

Understanding the full potential of speech as an input/output modality, covering the different guidelines and desirable adaptability features, in different application scenarios, is a complex, multivariate problem which often translates in a considerable development effort.

To tackle these issues we argue that an effort should be made to propose an architecture based on which a generic speech modality, decoupled from any particular application context, can be developed. This generic modality should encapsulate dealing with most of the complexity described above and should provide easier deployment of speech enabled systems.

The work presented in this paper is part of that effort and presents the methodology being followed to design and develop a module that enables speech interaction in applications. This methodology is characterized by the following notable aspects:

- A multimodal framework is considered and implemented;
- The speech modality is first developed as a generic modality and then integrated with the multimodal framework;
- Different application prototypes are used as a testbed, to inform development.

This article is organized as follows: Section 2 briefly presents background and related work; Section 3 describes our work regarding the proposal of a generic multimodal architecture supporting the development of generic modalities focusing the particular case of a generic speech modality; Section 4 presents two prototype applications which are used as part of our design and development pipeline for testing; finally, Section 5 presents some conclusions and ideas for further work.

2 Background and Related Work

Our work is aligned with recent W3C recommendations [10] for multimodal frameworks. This provides the grounds on which modalities are built, such as the speech modality presented in this paper. Therefore, to provide context, we briefly present the overall aspects of the multimodal framework, based on w3C recommendations, followed by an overview of relevant work presented in the literature regarding the use of speech in multimodal scenarios.

2.1 W3C Multimodal Framework

The W3C Recommendation [10] defines the major components of a multimodal system and identifies standard markup languages used to support communication between the components and data modules. The architecture can be divided into four major components (illustrated in Fig. 1):

- **Interaction Manager (IM)** – manages the different modalities. It is similar to the Controller in a Model View Controller (MVC) paradigm;
- **Modality Components** – representing input/output modules;
- **Runtime Framework** – acts as a container for all others, providing communication capabilities;
- **Data Component** – stores the data model.

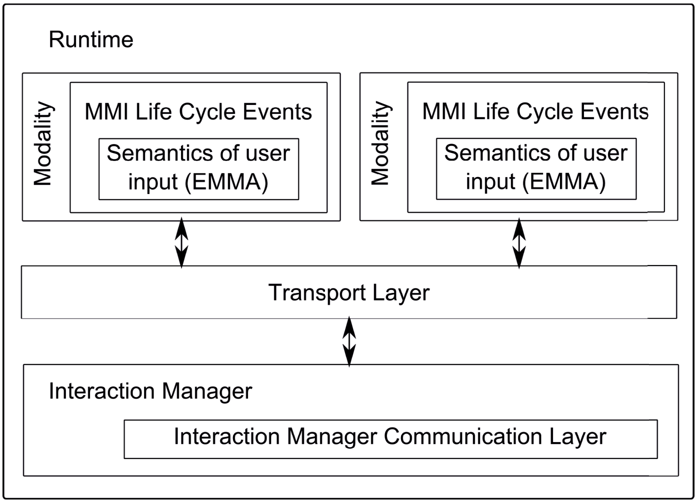
Runtime
Modality
MMI Life Cycle Events
Semantics of user input (EMMA)
Modality
MMI Life Cycle Events
Semantics of user input (EMMA)
Transport Layer
Interaction Manager
Interaction Manager Communication Layer

Fig. 1. The W3C Multimodal Architecture

Communication between Components (MMI Lifecycle Events). All communication is handled by MMI Lifecycle Events, a standard defined in the MMI Architecture. MMI Lifecycle events are messages exchanged between modalities and the Interaction manager, carrying the information of each event. Each message possesses common attributes. A request may possess attributes such as *context*, *source*, *target* or *requestID*. A response possesses attributes such as the *status*. Each MMI Life Cycle Event might also have the element *data* which is optional.

Standard Markup Language to Describe Events (EMMA). Extensible MultiModal Annotation markup language (EMMA) [4] is a standard language to describe events generated by different inputs, to be used within a multimodal system to exchange data information between inputs and multimodal components.

An EMMA document has three types of data:

- Instance data: Application-specific markup corresponding to input information;
- Data model: Constraints on structure and content of an instance;
- Metadata: Annotations associated with the data contained in the instance.

This language has a set of elements and attributes collected from the user's inputs, an *interpretation* element defines the event interpreted by the modality, with parameter such as *begin* and *end* time of the event, *confidence* of the recognition, *medium*, *mode* and recognized data.

SCXML. SCXML [5] is a markup language that defines a state chart machine and a data model. Its objective is to provide the application logics to the existing framework. The basic concepts of a state machine are states, transitions and events. When events occur, the machine tries to match the event to the transitions on the active state. If it matches, the target state is set as the new active state.

In SCXML, there are some extensions to a basic state machine. State machines can have executable content such as conditions, executable scripts, send messages to external entities or modalities and modify the data model. It also has two elements to execute content upon entering or exiting a state.

2.2 Speech for Interaction

Many recent applications using multimodal interaction explore the use of speech. It is one of the commonly present modalities in multimodal systems, appearing as part of the three most popular combinations mentioned by Bui et al. [11] for input: speech and lips movement, speech and gesture (including pen gesture, pointing gesture, human gesture) and speech, gesture, and facial expressions.

Popular combinations of output modalities, which include speech, are [11]: speech and graphics, speech and avatar and speech, text and graphics.

Adopting the definition of modality as "a way of exchanging information between humans [...] and machines, in some medium" [9], several "speech modalities" can be considered. In the Bernsen taxonomy three modalities are proposed, at atomic level: spoken discourse, spoken label-keywords and spoken notation [7].

The different Speech related modalities have different characteristics and, therefore, different suitabilities [7]. Spoken discourse is adequate for situated communication with the hearing and involving those who have the skills in interpreting and generating a particular language. It allows exchange of information when painstaking attention to detail is not required. If more complex data needs to be transmitted written language can be a better choice.

Spoken labels/keywords are suitable to convey small, isolated pieces of meaning as long as the context in which they are used helps reduce the inherent ambiguity. Bernsen et al. [8] refers the example of a user navigating a townscape. In that context, spoken words such as "house" or "door" are easily understood.

Spoken notation, might be a good option to convey information in the particular domain it refers too but, as it is often dynamic, it might be quite error prone or difficult to interpret by either human or machine [7] unless it is limited to particular contexts.

Speech is very resilient as a side channel, making it the ideal mode for "secondary task interfaces". These are interfaces for functions when the computational activity is not the primary task (ex: while driving) [13]. Furthermore, as discussed in Teixeira et al. [21], speech should not be used alone, it must be part of a multimodal input/output and, for some users or context of use (ex: mobile phone interaction with hands and eyes busy), will be the only useful modality.

The mTalk [17], developed by AT&T, Ford sync [1], Siri [2] and Xbox One [3] are well known examples of mutltimodal interaction that uses speech as a way to interact with the system, but those systems are commercial and closed solutions.

Mudra [16] and Manitou [14] are other examples of multimodal interaction frameworks that allow speech as a modality in the human-computer interaction. The first aims to process low-level streams and high level semantics and combine those events; the second aims for easy development of multimodal-enabled web applications.

3 Proposed Architecture for Speech Enabled Systems

Analysing existing work, it is important to note that most of the proposed solutions are very application oriented, i.e., the speech modality is developed tightly coupled with the envisaged application and device. As stressed before, we argue that this results in limited reuse of the developed modality, e.g., in a different application, yielding additional development costs and poses barriers, given the complexity of developing a speech modality, to its integration by third parties.

We propose a solution where modalities are decoupled and communicate with the applications through the multimodal framework enabling the reuse of modalities in other applications. Figure 2 illustrates one issue of current solutions and how it works for our proposed solution, namely, in the left we see that common scenarios use speech embedded as a part of the application and it is hard to reuse code to create new applications, on the other hand the desired scenario, on the right, has a speech modality decoupled from the application allowing the reuse of the modality in other applications.

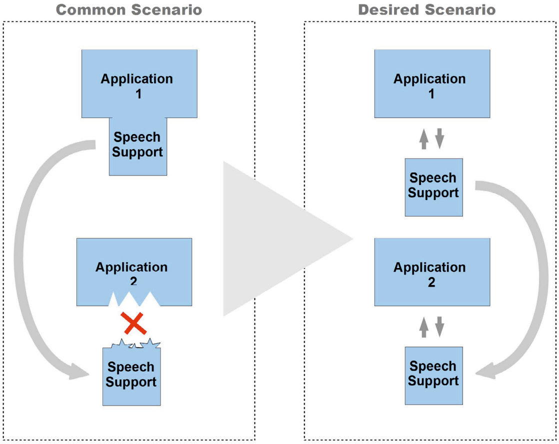

Fig. 2. Decoupled solution for the speech modality

3.1 Multimodal Framework

Our approach for speech enabled applications started by the development of a multimodal framework capable of managing different and generic modalities, supporting communication between modalities and the application.

The multimodal framework is directly based on the recommendations presented by the W3C, Multimodal Interaction (MMI) Architecture [10] and although it is focused on web scenarios, our goal is to extend it for interaction with mobile devices, tablets and AAL applications [23]. This choice is justified by the architecture's open standard nature and provides an answer to a significant part of the requirements presented, easing the creation and integration of new modules, as well as already existing tools.

Our multimodal framework has a main module, the Interaction Manager, which implements a state machine defined in SCXML that controls the flow of messages between modalities. To enable communication, the module implements an HTTP server listening to messages or requests sent by modalities, modalities only have to obey the message protocol in order to communicate with the system.

Therefore, having a standard for multimodal architecture helps application developers to avoid the unpractical situation of having to master each individual modality technology. This is particularly problematic as the number of technologies that can be used with multimodal interaction is increasing very fast. This standard architecture gives experts the possibility to develop standalone components [12] that can be used in a common way.

3.2 The Speech Modality

Considering the multimodal framework recommendations, modalities should be decoupled and communicate with the interaction manager with standard

MMI life cycle events, allowing other developers to focus on coding only the application.

Therefore, the proposed speech modality implements the communication languages described by the W3C architecture and communicates with the Interaction Manager which, in turn, communicates with the application sending the modalities' events.

The development of the speech modality starts with the creation of a generic modality supporting the different speech features required, considering both input and output. This modality is configured with a grammar, containing the possible sentences that the modality can recognize. We have created a tool that enables the translation of the grammars: by processing the grammar it generates all its possible sentences. Then, using translation services available on the web, each sentence is translated for the desired languages. Finally, the grammar is reassembled, creating a new grammar file for each language.

To support both mobile devices and desktop application, the modality has the capacity to process the recognition locally or remotely, enabling its use on mobile devices. When it is remotely, there is a local part of the modality to communicate with the remote part. Using this locally or remotely, does not affect how the framework is integrated. To accomplish this, services were created that process data and can be deployed in different locations (a device or a server).

Speech Recognition. The Asynchronous Speech Recognition (ASR) receives an audio stream with a spoken sentence, and the name of the grammar to be used to recognize the speech.

There are two kinds of grammars: GRXML, which is a W3C standard to specify the words or sentences to be recognized by the ASR, and ARPA, a statistical language model. The first type is more limited regarding the amount of sentences that can be recognized and is manually defined, but can return tags identifying the sentence's meaning. For ARPA, the creation of the grammar is automatic, since it is a statistical language model, but it requires large amounts of text in order to create the model, as well as the mechanisms to extract the meaning of the sentences.

Speech Synthesis. For this part of the service, called Text-to-Speech (TTS), the application sends a message with the information to be read to the user, the method to use to synthesize it to speech, using the Microsoft Speech platform (MSP), and the chosen voice. The service accepts other parameters such as speech volume and rate. The rate parameter defines the speed of the speech. Based on recent experiments in our group the default value chosen for the speed parameter makes the speech understandable for the elderly, and if the value increases, elderly people may have more difficulty in understanding it. The service returns an audio stream containing the spoken sentence.

3.3 Integration in the Multimodal Framework

In the second stage, the generic modality is integrated in a generic distributed multimodal framework, and dealt with as any other modality. Each modality follows the standard messaging specifications.

4 Application Prototypes

Finally, we have used the described multimodal architecture and speech modality to create two different applications, one targeting smartphones and other targeting home computers, with different use-case scenarios. These applications allow us to test and evaluate different aspects of our work informing further improvements to our proposed framework.

These applications, serving real application scenarios, are used as a test bed to improve our understanding of the different aspects involved, support brainstorming and inform development of future speech enabled applications.

Both applications use the Multimodal Framework and methodology previously discussed and each application targets a different device.

4.1 Newsreader

The application is a news reader developed for Windows 8, providing multimodal interaction for enhanced user experience and usability. It starts by loading some RSS news feeds from different sources depending on the users language and displaying the news to the user. At the same time, it processes the news contents to produce a list of headlines that it is used to configure a new grammar in the speech input modality.

An output modality called GUI, used as a part of the application, is continuously listening for messages coming from the Interaction Manager and it is responsible to update the interface of the application showing new content on the screen.

Figure 3 shows the modalities, states of the SCXML and the exchanged MMI Life Cycle events. Each modality, when it starts to run send a *NewContextRequest* to register in the Interaction Manager, it responds with a *NewContextResponse* informing if the registry was successful. After the speech modality recognizes the user sentence, it sends a *DoneNotification* with the event data to the Interaction Manager, which then sends a *StartResponse* to the GUI modality requesting some update in the user interface. The GUI modality replies with a *StartResponse* confirming the operation.

Different input modalities can be used to interact with the application. For instance, if the user wants to slide the container with the list of news, it can be done by any of the input modalities: via Kinect it is possible to swipe a hand to the left or right; Speech allows for actions to be active via words such as “left” or “right”; or Touch.

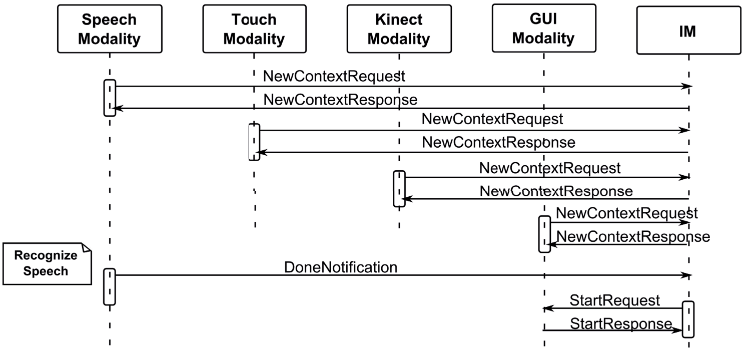

Fig. 3. Messages exchanged between the Interaction Manager (IM) and the modalities

In order for the user to read the entire body of the news, speech or touch can be used to select an article, by reading the headline or tapping the corresponding square.

Figure 4 presents an example of user interactions to read a particular article. The first screen shows the list of news by swiping the hand to the left or speak "left" the content slides to the left, it is shown in the screen in the upper right. Then the user says "Labours reputation at stake" to open the details of that article, as visible in the screenshot at the bottom left. Finally, the user says "go back" to return to the news list.

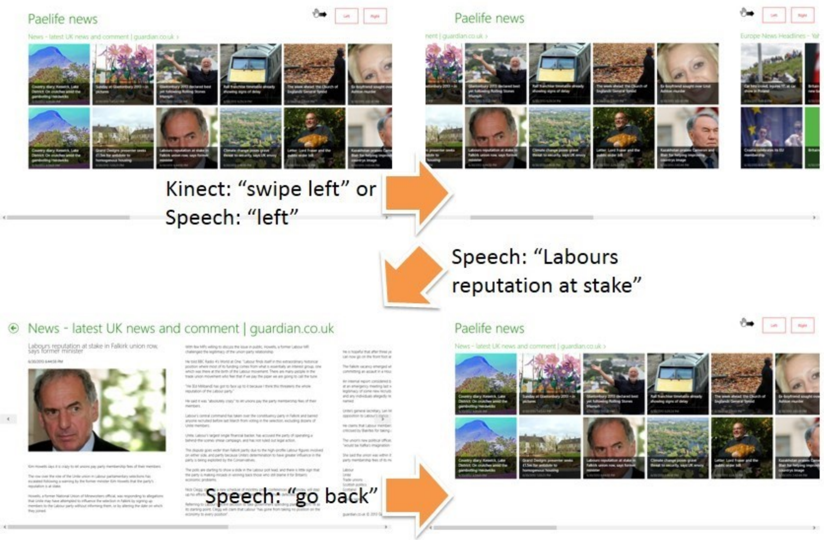

Fig. 4. Screens of the Newsreader application depicting some of the possible interactions

When an event occurs in the speech modality, the modality sends the event data to the IM to be processed. Upon processing it, the IM creates an action to be sent to an output modality, then the output modality presents that information to the user. Having a generic speech modality relieves the developers of having to handle with the recognizer, grammars, etc. In this scenario developers only have to inform about the sentences that can be recognized and a tag for which sentence.

4.2 Medication Assistant

This application, developed for Windows Phone, illustrated in Fig. 5, has two main functionalities: first, generating and showing medication intake alerts and, second, providing advice on how to proceed if the user misses a medication intake [24].

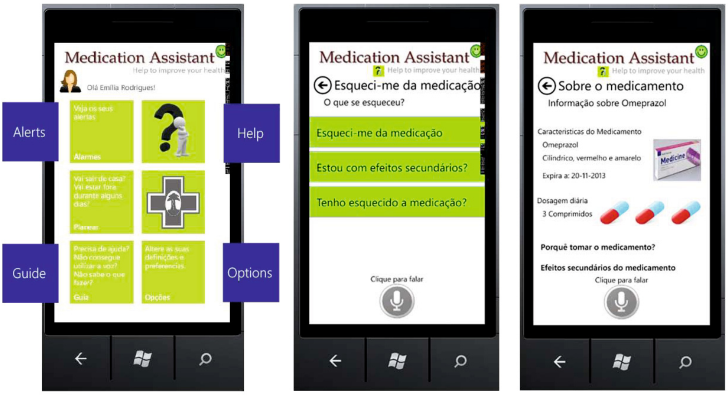
Medication Assistant
Help to improve your health
Olá Emília Rodrigues!
Alerts
Veja os seus alertas
Alarmes
Help
Vai sair de casa? Vai estar fora durante alguns dias?
Planear
Guide
Precisa de ajuda? Não consegue utilizar a voz? Não sabe o que fazer?
Guia
Options
Altere as suas definições e preferencias.
Opções
Medication Assistant
Help to improve your health
Esqueci-me da medicação
O que se esqueceu?
Esqueci-me da medicação
Estou com efeitos secundários?
Tenho esquecido a medicação?
Clique para falar
Medication Assistant
Help to improve your health
Sobre o medicamento
Informação sobre Omeprazol
Caracteristicas do Medicamento
Omeprazol
Cilindrico, vermelho e amarelo
Expira a: 20-11-2013
Medicine
Dosagem diária
3 Comprimidos
Porquê tomar o medicamento?
Efeitos secundários do medicamento
Clique para falar

Fig. 5. Graphical user interface of Medication Assistant depicting the main screen, advice on forgetting the intake of medicine and detailed information of medication

Moreover, the application provides additional information about the medications through multiple views making use of different representations (e.g. picture of the pills and the respective package, side effects, name, number of pills per day). The application implements two main use cases: "alert reading" and "missed medication intake". When the alert appears, the list of medications to take is displayed.

The user can interact with the application through speech or touch to obtain detailed information on each medication and in case he forgets to take the medication to inquire if he should take or not the medication. Speech can be used as a shortcut to go to specific views of the application, instead of having to select multiple options to select that view. In order to the system to give an efficient response it is necessary to provide relevant information to the system.

5 Conclusions

In this work, we propose a method to rapidly create new speech enabled application, by integrating the W3C multimodal framework and a generic speech modality in new application. To test our method we have developed two different application targeting different devices integrating the multimodal framework, serving as evidence of the increased ease of creating new and diversified application. Then, in a second stage, in which we are currently working on, this application allows us to define new requirements to enhance the generic modality.

Our method allows developers to easily implement an application with speech capabilities in multiple languages. Since the different modalities are decoupled from the application it is possible for the developers to focus only on the application features and design, and less concerns on the design of the interaction are required. Also, modalities can be extended to improve functionalities, to support other features, without the need to update the application. At time of writing the framework and modality is being explored for the development of Paelife Personal Assistan [22] and integrated multilingual support is being extended.

The decoupled nature of the interaction modalities and the existence of a standard multimodal framework pave the way to first attempts to consider multimodal design guidelines independently from the application, with the management of such aspects done at the multimodal framework level, e.g., regarding when to use speech, how to adapt the speech output considering the current context or how to use speech in parallel with other modalities.

Aknowledgments. The work presented is part of the COMPETE – Programa Operacional Factores de Competitividade and the European Union (FEDER) under projects AAL4ALL (www.aal4all.org): Part of the work presented was funded by FEDER, COMPETE and FCT in the context of AAL/0015/ 2009, IEETA Research Unit funding FCOMP-01-0124-FEDER-022682 (FCT-PEstC/ EEI/UI0127/2011) and project Cloud Thinking (QREN Mais Centro program, ref. CENTRO-07-ST24-FEDER-002031).

References

1. Ford sync, http://www.ford.com/technology/sync/
2. ios - siri, http://www.apple.com/ios/siri/
3. Xbox one, http://www.xbox.com/en-GB/xbox-one/meet-xbox-one
4. Baggia, P., Burnett, D.C., Carter, J., Dahl, D.A., McCobb, G., Raggett, D.: Emma: Extensible multimodal annotation markup language, http://www.w3.org/TR/emma/
5. Barnett, J., Akolkar, R., Auburn, R., Bodell, M., Burnett, D.C., Carter, J., McGlashan, S., Lager, T., Helbing, M., Hosn, R., Raman, T., Reifenrath, K., Rosenthal, N., Roxendal, J.: State Chart XML (SCXML): State Machine Notation for Control Abstraction, http://www.w3.org/TR/scxml/
6. Bernsen: Towards a tool for predicting speech functionality. Speech 23, 181–210 (1997)
7. Bernsen, N., Dybkjaer, L.: Multimodal Usability (2009)

8. Bernsen, N.O.: Multimodal usability: More on modalities (December 2012), http://www.multimodalusability.dk/
9. Bernsen, N.O.: Multimodality in language and speech systems – from theory to design support tool. In: Granstrm, B., House, D., Karlsson, I. (eds.) Multimodality in Language and Speech Systems, Text, Speech and Language Technology, vol. 19, pp. 93–148. Springer, Netherlands (2002)
10. Bodell, M., Dahl, D., Kliche, I., Larson, J., Porter, B.: Multimodal Architecture and Interfaces, W3C (2012), http://www.w3.org/TR/mmi-arch/
11. Bui, T.H.: Multimodal dialogue management - state of the art. Technical Report TR-CTIT-06-01, Centre for Telematics and Information Technology University of Twente, Enschede (January 2006)
12. Dahl, D.A.: The W3C multimodal architecture and interfaces standard. Journal on Multimodal User Interfaces (April 2013), http://link.springer.com/10.1007/s12193-013-0120-5
13. Deketelaere, S., Cavalcante, R., RasaminJanahary, J.F.: Oasis speech-based interaction module. Tech. rep. (2009)
14. Hak, R., Dolezal, J., Zeman, T.: Manitou: A multimodal interaction platform. In: 2012 5th Joint IFIP Wireless and Mobile Networking Conference (WMNC), pp. 60–63 (September 2012)
15. Hale, K.S., Reeves, L., Stanney, K.M.: Design of systems for improved human interaction (2011)
16. Hoste, L., Dumas, B., Signer, B.: Mudra: A unified multimodal interaction framework. In: Proceedings of the 13th International Conference on Multimodal Interfaces, ICMI 2011, pp. 97–104. ACM, New York (2011)
17. Johnston, M., Fabbrizio, G.D., Urbanek, S.: mtalk - A multimodal browser for mobile services. In: INTERSPEECH, pp. 3261–3264. ISCA (2011)
18. Nass, C., Brave, S.: Wired for Speech: How Voice Activates and Advances the Human-computer Relationship. MIT Press (2007)
19. Sarter, N.: Multimodal information presentation in support of human-automation communication and coordination, vol. 2, pp. 13–35. Emerald Group Publishing Limited (2002)
20. Sarter, N.B.: Multimodal information presentation: Design guidance and research challenges. International Journal of Industrial Ergonomics 36(5), 439–445 (2006)
21. Teixeira, A., Braga, D., Coelho, L., Fonseca, J., Alvarelhão, J., Martins, I., Queirós, A., Rocha, N., Calado, A., Dias, M.: Speech as the basic interface for assistive technology. In: Proc. 2th International Conference on Software Development for Enhancing Accessibility and Fighting Info-Exclusion, DSAI (2009)
22. Teixeira, A., Hämäläinen, A., Avelar, J., Almeida, N., Németh, G., Fegyó, T., Zainkó, C., Csapó, T., Tóth, B., Oliveira, A., Dias, M.S.: Speech-centric multimodal interaction for easy-to-access online services – A personal life assistant for the elderly. In: Proc. DSAI 2013, Procedia Computer Science (November 2013)
23. Teixeira, A.J.S., Almeida, N., Pereira, C., Silva, M.O.: W3c mmi architecture as a basis for enhanced interaction for ambient assisted living. In: Get Smart: Smart Homes, Cars, Devices and the Web, W3C Workshop on Rich Multimodal Application Development. New York Metropolitan Area, US (July 2013)
24. Teixeira, A.J.S., Ferreira, F., Almeida, N., Rosa, A.F., Casimiro, J., Silva, S., Queirós, A., Oliveira, A.: Multimodality and adaptation for an enhanced mobile medication assistant for the elderly. In: Third Mobile Accessibility Workshop (MOBACC), CHI 2013 Extended Abstracts (April 2013)