Enhanced Interaction for the Elderly supported by the W3C Multimodal Architecture

Nuno Almeida António Teixeira
Dep. Electronics Telec & Informatics/IEETA, Universidade de Aveiro
Aveiro, Portugal
{nunoalmeida, ajst}@ua.pt

Abstract

Some elderly have resistance in what concerns the use of technology. Improving usability and accessibility is expected to increase their acceptance and use of new technologies. Combination of different modalities to support enhanced interaction, particularly including the use of spoken language, is one path with recognized potential to enhance usability and accessibility. Despite the recognized potential, there is not much work on creating the conditions for simple and fast development of such forms of interaction.

In this paper we propose the adoption of the recent W3C multimodal architecture as the basis to create enhanced multimodal interaction for the Elderly, describing its general architecture and how we implemented a multimodal framework. The implementation is composed by several components, being one of the most important, the Interaction Manager, responsible for receiving event messages from the input modalities and making decisions on how to process those messages. It is also described the implementation of the modalities used in this context. As a proof-of-concept, the paper ends with the presentation of an application to provide access to news feeds by voice and body gestures, used to test the developed framework.

Keywords

Multimodal framework, interaction, natural language, speech technologies, elderly

1. INTRODUCTION

The elderly show some resistance in adopting technology [Park10], depriving them from the benefits it has to offer. This problem is gaining more importance, since we live longer, and are likely to be physically, socially and cognitively active until older ages.

We need to fight isolation and exclusion to allow the elderly to be more productive, independent and to have a more social and fulfilling life. This can be done by improving the accessibility to existing and new devices and services. All this should be made possible at people's homes, since elderly people have sometimes some level of impairments caused by age, reducing their mobility.

In this context, our aim is to build a multimodal framework to make easier to create multimodal applications, and, enhancing usability. It's specially targeted for elderly but it can also be used to develop applications for other groups of users. For the time being, the main areas of application are Ambient Assisted Living (AAL) and Personal Assistants for Social Interaction. The second is directly related to project AAL PaeLife [Paelife11].

In order to design such framework we adopt from the beginning the AMITUDE model proposed in [Bernsen09]. It is a generic model of the aspects involved when someone uses a multimodal system, providing designers with a conceptual development-for-usability framework that describes all aspects of system that must be taken into account when developing for multimodal usability. Using this model we found the aspects of interaction to consider. For example, the Personas of the PaeLife project are older adults aged more than 60 years who have some degree of experience with computers, although they may not be proficient using them. Taking in consideration these Personas, traditional interfaces should be avoided, as they may not be familiar to users and they are not easy-to-use.

PaeLife Personas, while suffering from typical age-related ailments (reduced dexterity, etc.), do not present any serious condition that could heavily compromise their interaction with computers. Therefore, physiological interfaces are not suitable for our project, since they are cumbersome, intrusive and not easy to use.

This method also helped in requirements analysis, showing that the multimodal framework should support multiple devices, such as a Home Computer connected to the TV, since the Personas spend most of its time at home; and a tablet or smartphone, in to support liberty of movement at home. The user can interact with the TV or tablet separately or combining the both. We have concluded that speech as an input modality is an important feature for interaction. The other modalities found more relevant for the multimodal framework that would enhance user usability were: touch, gestures and the ones integrating classical GUI.

The next section presents the main requirements of the multimodal framework; the third section presents a W3C standard for use of multimodal interaction. In the fourth section we describe our implementation of a Multimodal Framework and some modalities that are included in the framework. In the fifth section we describe a Demo application integrating the Multimodal Framework.

2. MAIN REQUIREMENTS

The framework aims at integrating developments regarding interaction modalities to make easier inclusion of multimodal interaction into future applications. The framework should the follow set of main requirements:

- Support to multiple modalities (input and output);
- Support for distribution of modalities across computational devices (PCs, tablets, etc.);
- Loosely coupled architecture;
- Extensibility;
- Adoption of international standards, avoiding as much as possible proprietary or closed solutions;
- Flexibility, mainly by providing the possibility to change or add modules without the rest of the system acknowledge that;
- Clean and easy to use.

3. MULTIMODAL INTERACTION ARCHITECTURE

The developed multimodal framework is directly based on the "standards" defined by the W3C, Multimodal Interaction (MMI) Architecture [Bodell12]. This choice is justified by the architecture's open standard nature. This architecture 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.

Having a standard for multimodal architecture helps avoiding the unpractical situation for application developers of needing 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 [Dahl13] that can be used in a common way.

The W3C's Recommendation [Bodell12] 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 Figure 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.

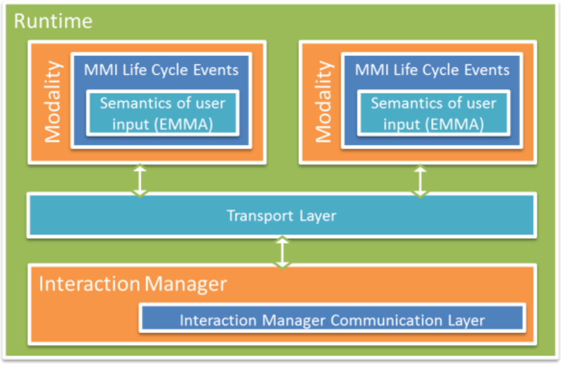
Runtime
Modality
MMI Life Cycle Events
Semantics of user input (EMMA)
Modality
MMI Life Cycle Events
Semantics of user input (EMMA)
Transport Layer
Interaction Manager
Interaction Manager Communication Layer

Figure 1 - The W3C Multimodal Architecture

Other components can be considered in the architecture, such as fusion components and fission components or registration and discovery of modalities. Conceptually fusion and fission components can be part of the IM or implemented as complex modalities components [Dahl13], making use of the fractal natural of the architecture that allows several levels of interaction managers. These aspects still in discussion inside the W3C workgroup [Teixeira13].

3.1 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.

3.2 Standard Markup Language to Describe Events (EMMA)

EMMA (Extensible MultiModal Annotation markup language) [Baggia09] 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.

4. FRAMEWORK IMPLEMENTATION

4.1 Runtime

To assure communication, each module should include its own HTTP server. Either on the IM or modality components, the server is responsible for receiving/sending the messages. However, polling might also be used on simpler modalities. Using the described standards MMI and EMMA, the IM implements an HTTP server, it receives MMI Life Cycle Events from modalities, and has the possibility to respond to that channel.

4.2 Interaction Manager

It starts by loading a SCXML file - see next section - and creating a HTTP server capable of receiving MMI lifecycle events. When a MMI lifecycle event is received by the server, the IM parses the message and sends it to the core module, the state machine triggers an event. However it can also include modifications on the data model, or sending a new MMI lifecycle event to others modalities.

4.2.1 SCXML

SCXML [Barnett12] 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 occurs, 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 conditions; executable scripts; send messages to external entities or modalities; modify the data model. It also has two elements to execute content upon entering or exiting a state.

4.3 Modalities

The modalities created to include in the developed multimodal framework are based in technologies available from previous research and development by the authors and by partners of the project. In the current framework are included body gestures and speech input and output. Other technologies are being developed to create new modalities regarding Natural Language Understanding, Natural Language Generation and Touch.

4.3.1 Modalities for Speech

The speech input modality was created using the Microsoft Speech Platform [Msft13] in C#. The modality requires a grammar that defines recognition sentences. The grammar however is not included in the module and must be sent by the associated application (the IM), making use of this modality more general. Grammars follow a predetermined standard, GRXML that is a W3C standard [Hunt04] markup language that defines a grammar structure for speech recognition containing information of words or sentences that the engine should be aware.

The configuration of the Speech input modality is capable of using the language packs in development in PaeLife, providing support to multilingual speech inputs.

4.3.2 Input Modality for Body Gestures

The body gestures modality uses the PaeLife Kinect Framework built for the PaeLife project for recognition of user gestures. It supports two gestures: Swipe Left and Swipe Right. The framework uses the Microsoft Kinect SDK to track the user skeleton and by analyzing skeleton points for both hands, it recognizes swipe gestures.

4.3.3 Graphical output

It acts as an adapter for a concrete graphical application. Upon receiving an MMI Life Cycle event, the module may call a method within the application to change some aspects of the visual interface. For instances, a MMI Life Cycle Event with the event “Swipe Left”, calls a method to make the displayed content to slide.

5. PROOF-OF-CONCEPT APPLICATION

To serve as a test, a demo application was created. The application recreates a news reader, adding a multimodal interaction for enhanced user experience and usability.

Upon start, it loads some RSS news feeds, displaying the news to the user. At the same time, it processes the news contents to produce a list of headlines it uses to configure the speech input modality grammars.

The Graphical output modality (part of the application) is continuously listening for messages coming from the IM. The Figure 2 shows the modalities, states of the SCXML and the exchanged MMI Life Cycle events.

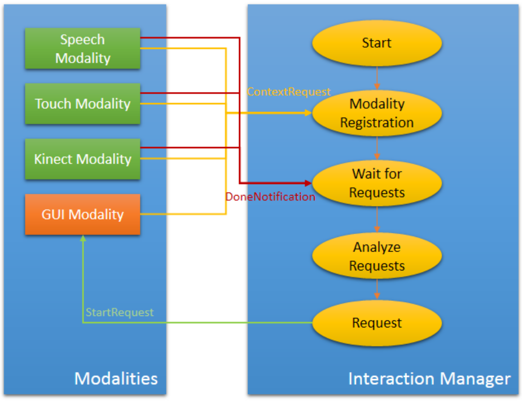

Figure 2 - Exchanged between IM and the modalities

If an event occurs in the body gestures modality, the modality sends it to the IM to be processed. Upon processing it, the IM creates an event to be sent to the Graphical output modality, to perform changes in the interface. Constantly, the application communicates with the Speech Modality to inform of sentences that can be recognized.

Each modality allows the user to interact with the application. For instance, to slide the container with the list of news, any modality of the input modalities can be used: 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 (in the process of creation of a modality);

To read details of the news, the speech or touch modality can be used, by reading the headline or tapping the square corresponding to the news. The Figure 3 represents an interaction from the users to read news.

Currently, each time a modality sends an event to the IM, it sends a message to the Graphical output modality to change the displayed information.

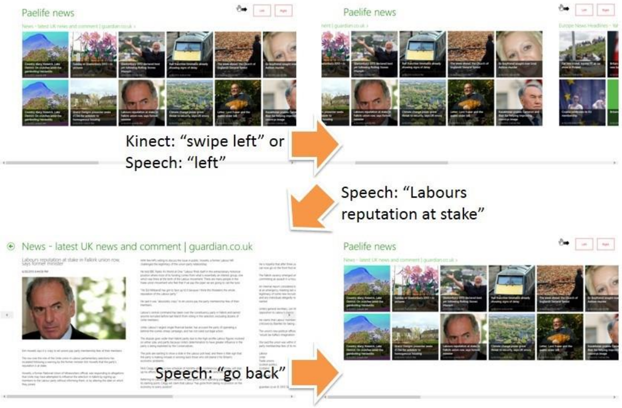

Figure 3 - Screens of the application and interaction

6. CONCLUSION

The developed multimodal framework provides the tools for developers for faster creation of multimodal applications. It also makes much easier the addition of new modalities to the system. It's possible to use different modalities and make this transparent to the application since the Interaction Manager controls the flow of messages and the information sent to the application can be unified across modalities. New modalities also only have to integrate with the IM.

The developed architecture allows the use of modalities with adaptation and configuration mechanisms: it's possible to send a message to the Speech Modality to change the current recognized language as the information about the interpretation of the recognized sentence sent to the IM is the same despite the used language. In our context, the European PaeLife project with the objective of supporting several languages (Portuguese, French, Polish, Hungarian, and English), is very important the abstraction of the used language.

Once the communication between modules is done with HTTP protocol, modalities can be created in different platforms, simplifying the creation of systems using different devices, such as a Home Computer and Tablet that provide a joint interaction experience.

6.1 Future work

In the near future, our intention is to make an evaluation off the framework with different real users. This evaluation will help to improve the framework and the modalities.

We are also working in having more advanced modalities, such as a Speech Modality with the capability of translate grammars to other languages, and inclusion of techniques to extract semantic information. The IM or a fission module will choose a suitable output modality to present information to the user (Graphical output, TTS or both).

This architecture will also support the Personal Life Assistant (PLA) in development by the partners of the PaeLife project [Saldanha13].

7. ACKNOWLEDGMENTS

Authors acknowledge the funding from AAL JP and national agencies: in the context of AAL/0015/2009 and IEETA Research Unit funding FCOMP-01-0124-FEDER-022682 (FCT-PEstC/EEI/UI0127/2011).

8. REFERENCES

[Baggia09] P. Baggia, D. C. Burnett, J. Carter, D. A. Dahl, G. McCobb e D. Raggett, "EMMA: Extensible MultiModal Annotation markup language," W3C, 2009. [Online]. Available: http://www.w3.org/TR/emma/. [Acedido em 2013].

[Barnett12] J. Barnett, R. Akolkar, RJ Auburn, M. Bodell, D. C. Burnett, J. Carter, S. McGlashan, T. Lager, et. Al. "State Chart XML (SCXML): State Machine Notation for Control Abstraction," [Online]. Available: http://www.w3.org/TR/scxml/

[Bernsen09] Bernsen, N. O., & Dybkjær, L. *Multimodal usability*. Berlin ; New York: Spring, 2009

[Bodell12] M. Bodell, D. Dahl, I. Kliche, J. Larson, B. Porter, et al. Multimodal Architecture and Interfaces, W3C, 2012. [Online]. Available: http://www.w3.org/TR/mmi-arch/. [Acessed in 2013].

[Dahl13] A Dahl, Deborah. The W3C multimodal architecture and interfaces standard. *J Journal on Multimodal User Interfaces. Springer-Verlag* April 2013

[Hunt04]A. Hunt, S. McGlashan. "Speech Recognition Grammar Specification Version 1.0", W3C, 2004 [Online]. http://www.w3.org/TR/speech-grammar/

[Msft13] "Microsoft Speech Platform," Microsoft, 2013. [Online]. Available: http://msdn.microsoft.com/en-us/library/hh361572.aspx. [Acedido em 2013]

[Paelife11] PaeLife Project (2011-2014), www.paelife.eu

[Park10] Park, Sung; Fisk, Arthur D.; Rogers, Wendy A. Human Factors Consideration for the Design of Collaborative Machine Assistants. *Handbook of Ambient Intelligence and Smart Environments*. Springer-Verlag US, 2010, p. 96

[Saldanha13] N. Saldanha, J. Avelar, M. Dias, A. Teixeira et al. A Personal Life Assistant for "natural" social interaction*: the PaeLife project*, AAL Forum, 2013.

[Teixeira13] A. Teixeira, N. Almeida, C. Pereira, M. O. Silva. W3C MMI Architecture as a Basis for Enhanced Interaction for Ambient Assisted Living. Get Smart: *Smart Homes, Cars, Devices and the Web, W3C Workshop on Rich Multimodal Application Development*, New York Metropolitan Area, US, July 2013