Universidade de Aveiro Departamento de
2017 Eletrónica, Telecomunicações e Informática

Nuno Filipe Correia de Almeida

Interacção multimodal: contribuições para simplificar o desenvolvimento de aplicações

Multimodal interaction: contributions to simplify application development

Universidade de Aveiro
2017
Departamento de
Eletrónica, Telecomunicações e Informática

Nuno Filipe Correia de Almeida

Interacção multimodal: contribuições para simplificar o desenvolvimento de aplicações

Multimodal interaction: contributions to simplify application development

Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Informática, realizada sob a orientação científica do Doutor António Joaquim da Silva Teixeira, Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e do Doutor Samuel de Sousa Silva, Investigador do Instituto de Engenharia Electrónica e Informática de Aveiro

o júri

presidente

Prof. Doutora Maria Herminia Deulonder Correia Amado Laurel
professora catedrática da Universidade de Aveiro

Prof. Doutor Daniel Jorge Viegas Gonçalves
professor associado do Instituto Superior Técnico da Universidade de Lisboa

Prof. Doutor António Joaquim Silva Teixeira
professor associado da Universidade de Aveiro (orientador)

Prof. Doutor Hugo Alexandre Paredes Guedes Silva
professor auxiliar com agregação da Escola de Ciências e Tecnologia, Universidade de Trás-os-Montes e Alto Doutro

Prof. Doutor Carlos Jorge da Conceição Teixeira
professor auxiliar da Faculdade de Ciências, Universidade de Lisboa

Prof. Doutor Miguel Augusto Mendes de Oliveira Silva
professor auxiliar da Universidade de Aveiro

agradecimentos

Em primeiro lugar, quero agradecer aos meus orientadores, Professor Doutor António Teixeira e Doutor Samuel Silva, por todo o empenho e apoio disponibilizados ao longo dos trabalhos realizados para esta tese.

Aos meus amigos e colegas que me ajudaram e apoiaram nos momentos mais difíceis.

Por último, quero dedicar este trabalho às pessoas mais importantes da minha vida, a minha família. Os meus sinceros agradecimentos, aos meus pais e aos meus irmãos por acreditarem em mim.

palavras-chave Interacção, interacção multimodal, multiplos dispositivos, fala, gestos, fusão

resumo

A forma como interagimos com os dispositivos que nos rodeiam, no nosso dia-a-dia, está a mudar constantemente, consequência do aparecimento de novas tecnologias e métodos que proporcionam melhores e mais aliciantes formas de interagir com as aplicações. No entanto, a integração destas tecnologias, para possibilitar a sua utilização alargada, coloca desafios significativos e requer, da parte de quem desenvolve, um conhecimento alargado das tecnologias envolvidas. Apesar de a literatura mais recente apresentar alguns avanços no suporte ao desenho e desenvolvimento de sistemas interactivos multimodais, vários aspectos chave têm ainda de ser resolvidos para que se atinja o seu real potencial. Entre estes aspectos, um exemplo relevante é o da dificuldade em desenvolver e integrar múltiplas modalidades de interacção.

Neste trabalho, propomos, desenhamos e implementamos uma framework que permite um mais fácil desenvolvimento de interacção multimodal. A nossa proposta mantém as modalidades de interacção completamente separadas da aplicação, permitindo um desenvolvimento, independente de cada uma das partes. A framework proposta já inclui um conjunto de modalidades genéricas e módulos que podem ser usados em novas aplicações. De entre as modalidades genéricas, a modalidade de voz mereceu particular atenção, tendo em conta a relevância crescente da interacção por voz, por exemplo em cenários como AAL, e a complexidade associada ao seu desenvolvimento. Adicionalmente, a nossa proposta contempla ainda o suporte à gestão de aplicações multi-dispositivo e inclui um método e respectivo módulo para criar fusão entre eventos.

O desenvolvimento da arquitectura e da framework ocorreu num contexto de I&D diversificado, incluindo vários projectos, cenários de aplicação e parceiros internacionais. A framework permitiu o desenho e desenvolvimento de um conjunto alargado de aplicações multimodais, sendo um exemplo digno de nota o assistente pessoal AALFred, do projecto PaeLife. Estas aplicações, por sua vez, serviram um contínuo melhoramento da framework, suportando a recolha iterativa de novos requisitos, e permitido demonstrar a sua versatilidade e capacidades.

keywords interaction, multimodal interaction, multi-device, speech, gestures, fusion

abstract

The way we interact with the devices around us, in everyday life, is constantly changing, boosted by emerging technologies and methods, providing better and more engaging ways to interact with applications. Nevertheless, the integration with these technologies, to enable their widespread use in current systems, presents a notable challenge and requires considerable knowhow from developers. While the recent literature has made some advances in supporting the design and development of multimodal interactive systems, several key aspects have yet to be addressed to enable its full potential. Among these, a relevant example is the difficulty to develop and integrate multiple interaction modalities.

In this work, we propose, design and implement a framework enabling easier development of multimodal interaction. Our proposal fully decouples the interaction modalities from the application, allowing the separate development of each part. The proposed framework already includes a set of generic modalities and modules ready to be used in novel applications. Among the proposed generic modalities, the speech modality deserved particular attention, attending to the increasing relevance of speech interaction, for example in scenarios such as AAL, and the complexity behind its development. Additionally, our proposal also tackles the support for managing multi-device applications and includes a method and corresponding module to create fusion of events.

The development of the architecture and framework profited from a rich R&D context including several projects, scenarios, and international partners. The framework successfully supported the design and development of a wide set of multimodal applications, a notable example being AALFred, the personal assistant of project PaeLife. These applications, in turn, served the continuous improvement of the framework by supporting the iterative collection of novel requirements, enabling the proposed framework to show its versatility and potential.

Index

List of Figures

List of Tables

Acronyms

AAL	Ambient Assisted Living
API	Application Programming Interface
ASR	Automatic Speech Recognition
CARE	Complementarity, Assignment, Redundancy and Equivalence
CASE	Concurrent, Alternate, Synergistic, Exclusive
EMMA	Extensible MultiModal Annotation markup language
GRXML	Speech Recognition Grammar Specification
GUI	Graphical User Interface
HCI	Human-Computer interaction
HMM	Hidden Markov Models
HTS	HMM-based speech synthesis system
HTTP	Hypertext Transfer Protocol
ICF-US US	International Classification of Functioning based Usability Scale
IM	Interaction Manager
MMI	Multimodal Interaction
NLG	Natural Language Generator
NLU	Natural Language Understanding
PC	Personal computer
RSSI	Received Signal Strength Indication
SAPI	Microsoft Speech API
SCXML	State Chart XML

SDK	Software Development Kit
SKEMMI	Sketch Multimodal Interactions
SLU	Spoken Language Understanding
SMUIML	Synchronized Multimodal User Interfaces Modelling Language
SSDP	Simple Service Discovery Protocol
SSML	Speech Synthesis Markup Language
SUS	System Usability Scale
TTS	Text to Speech
TV	Television
UPnP	Universal Plug and Play
W3C	World Wide Web Consortium
WIMP	Windows, Icons, Menus, Pointer

Chapter 1

Introduction

Humans interact in a rich way with others through the different senses (Norris, 2004) represented in Figure 1-1, enabling them to absorb the information, so that it can be interpreted and transmit information to others. For instance, in a human-to-human conversation someone speaks and makes gestures to transmit a message while the other uses the senses of hearing and sight to capture and interpret that message.

The same way humans communicate and interact with each other, these senses and mechanisms might be used for interacting with computers. Human-to-human communication provides a natural felling of interaction, where interaction with a computer somehow does not, with the main ways of interaction nowadays being mostly oriented to mouse and keyboard with a rising trend for touch interfaces.

Figure 1-1 – Five of the human senses (sight, hearing, smell, taste, touch)

Every day we are a step closer to accomplish more natural and intuitive ways of interaction with machines. Evolving technologies are allowing, as of today, to accomplish what some years ago were scenes of sci-fi movies, such as having a personal electronic assistant. This evolution will continue to bring us new and increasingly evolved technologies that are very intuitive and fun, such as, for instance, in the movie Avengers (Figure 1-2), the interaction of Tony Stark, the Iron Man, with virtual 3D objects or speaking with a very intelligent assistant.

1.1 Motivation

The way users interact with applications is changing, boosted by a widespread availability of new devices (Figure 1-3) and technologies that enable a richer set of options for interaction. Within AAL's objectives (Moschetti, Fiorini, Aquilano, Cavallo, & Dario, 2014), multimodality may enable shortening the gap between the user and the ambient. Providing different interaction methods, not only serves creating redundancy, but also potentially increases usability rates and acceptance.

Figure 1-2 - Image of Tony Stark (Iron Man) interacting with a hologram, from the Avengers Movie, Joss Whedon (2012), Marvel Studios and Paramount Pictures

Figure 1-3 - Devices for Interaction. From left, display, VR headset, speakers, touch display, fitness tracker, Kinect, microfone.

At the same time, in the context of human computer interaction, and in light of current technologies and how they enter our daily lives, there is an increasing demand for more natural ways of interaction, in line with human to human communication, such as speech or gestures. Speech has a strong potential in areas such as robot assistants (Teixeira, 2014) or assistive technologies (Teixeira, Braga, Coelho, & Fonseca, 2009), particularly by allowing hands-free interaction and benefiting from its intuitive use, potentially requiring minimal learning. Use of hands and body gestures can also be very useful, but without

the benefits of hands-free interaction and often including a need for additional learning and training.

Ambient Assisted Living (AAL) is increasingly relevant in our ageing society. Its role in providing the means for people to retain their autonomy and quality of life for a much longer time, while living at their homes, with continuous access to services supporting them on their physical and cognitive difficulties, is well recognized (Steg, Strese, Loroff, Hull, & Sophie Schmidt, 2006). And the impact goes far beyond improving life quality by also enabling a better management and deployment of healthcare resources. Furthermore, as many Internet services are now part of everyday life of technological non-impaired people, one can envision AAL services as also playing an important role in the daily lives of every person.

1.2 Challenges and Problems

The creation of AAL applications and services, particularly if considering pervasive scenarios is not without its challenges. In recent years we have seen applications that provide multiple interaction features (e.g., speech or touch), **but the support for multiple modalities in a truly multimodal setting entails a great development effort and the need to have a detailed knowledge regarding the different technologies needed to integrate each modality**.

Furthermore, different devices (smartphones, tablets, game boxes) potentially offer a diverse range of input and output modalities (e.g., speech, touch, gestures, and graphics) and it is important to devise how we should address this diversity. The option of developing for each particular device is a possibility, although easily identified as a limitation since, beyond an exponential development effort, it would preclude the required adaptability to different users and contexts. **If a new modality is available, it is desirable that it can be easily recognized and integrated into the system. The same is expected for new devices or sensors.**

The multitude of available devices, e.g., in a domestic scenario, such as smartphones, smart watches, tablets, smart TVs, and media centers, make us question how we could take advantage of the device set as a whole and not just as single devices. It seems relevant that **the best features of each device, whether a large screen, a touch pad, or speech output, should be used interchangeably, or in parallel, to best serve the context or task at hand**.

A number of multimodal frameworks are available, some lack important features such as extensibility and decoupling, and each one uses a different architecture. In some cases, the communication protocols are a barrier to implement in other device's architectures. Among the different efforts presented in the literature, the W3C multimodal architecture addresses important requirements to develop MultiModal Interaction (MMI), but there are several open issues that still do not allow developers to harness the full potential of multimodality and, consequently, result in MMI that can still move ahead on adaptability, ease of use and user engagement.

This work looks into these different challenges and desirable features for AAL and surveys the literature for relevant support for their design and development, particularly focusing interaction in a multimodal, multi-device AAL scenario. The main question concerns **if current technologies and methods provide adequate and systematic support for designing and developing systems that fully harness the available interaction technology, enabling the evolution of AAL towards its improved and increasingly relevant role in people's daily lives.**

In summary, the problems are:

- Hard to develop and support multiple modalities
    - Very different/complex modality technologies
    - Modalities are typically designed tightly coupled with applications
- Many frameworks with different approaches

    - Diverse architectures and communication protocols
    - Components that are not interchangeable
- No generic tools to develop multimodal applications
- No readily available modalities even for proposals such as the W3C MMI architecture standard
    - Lack of initial support
    - Prevents developers from adopting the architecture

1.3 Objetives

Considering the different aspects discussed previously, regarding the different challenges in designing and developing multimodal applications, and how the state-of-the-art is still lacking proper answers, we consider that a novel approach is essential. Therefore, **we aim to address these issues proposing a multimodal framework that supports context awareness and easy instantiation of MMI applications with a particular emphasis in AAL scenarios.** It should provide clear design definitions and allow increased compatibility between the developed applications and present and future input/output modalities.

The proposed framework should consider the following notable requirements:

- Have a usable implementation of an MMI architecture capable of supporting development of multiple applications;
- Propose a methodology for easier creation of MMI applications based on the framework;
- Provide generic modalities (i.e., developed independently, decoupled from the applications) with adaptation capabilities;
    - To make the use of the proposed architecture faster and simpler

    - Provide support for multiple languages for all speech related modalities and, as much as technology allows, automatically manage the support for multilingual content based on the original content.
- Support for multi-device applications that can also be standalone;
- Propose a module for fusion of events generated by input modalities;
- Address additional issues endowing the proposed framework with, for example, support for registration and discovery of the modalities and management of unknown modalities;

The work to be carried out should be supported by continuous validation of the proposed features by using, in a first stage, small prototype applications and, as the work matures, real application scenarios provided by ongoing work on projects such as IRIS[1]. These, not only provide a real application context, but also joint development by several partners, a scenario where the proposed work can show its mettle.

The work carried out on this thesis follows the Engineering Design Method (Dieter & Schmidt, 2013). The method can be summarized in a few stages. After defining the problem, the work starts by researching, observing and analyzing current state of the art solutions, followed by the specification of the requirements for the product. Given the requirements, solutions are conceptualized. These are then developed and tested. This process is repeated until a good degree of satisfaction is attained based on the extent to which the proposed solutions address the identified problems.

[1] http://www.iris-interaction.eu/

1.4 Contributions

Over time, the relevance of the work described in this thesis, with clear contributions to the state-of-the art, allowed the publication of its main outcomes in multiple peer-reviewed forums. Some of the publications address contributions mainly performed the author of this work, but others resulted from the collaborative effort of a research team. In the following paragraphs a brief contextualization of some selected publications, deemed representative of the main contributions, is presented.

One of the main objectives of this work was the definition of a **multimodal architecture**, identification of the main modules and its development. The main outcomes, relevant for this context, were published in:

- *Nuno Almeida, Samuel Silva, António Teixeira, Diogo Vieira. Multi-Device Applications Using the Multimodal Architecture. In D. Dahl (Ed.), Multimodal Interaction with W3C Standards - Toward Natural User (1st ed., pp. 367–383). Springer International Publishing, 2017.*

  This book chapter presents the most advanced version of the proposed multimodal architecture.

- *António Teixeira, Nuno Almeida, Carlos Pereira, Miguel Oliveira e Silva, Diogo Vieira, Samuel Silva. Applications of the Multimodal Interaction Architecture in Ambient Assisted Living. In D. Dahl (Ed.), Multimodal Interaction with W3C Standards - Toward Natural User (1st ed., pp. 271–291). Springer International Publishing, 2017.*

  This is a book chapter, published in the first book focusing the community efforts aligned with the W3C standard approach to Multimodal Interaction. It presents our AAL

scenarios and the proposed multimodal architecture, which started by the initial proposal of the W3C standard.

- *António Teixeira, Nuno Almeida, Carlos Pereira, Miguel Oliveira e 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*

  The publication describes the vision of our research group about the use of multimodal interaction in AAL application.

- *António Teixeira, Nuno Almeida, Carlos Pereira, Miguel Oliveira e Silva, José Casimiro Pereira. Serviços de Suporte à Interacção Multimodal. Laboratório Vivo de Usabilidade (Living Usability Lab), ARC Publishing, p. 151-165, October 2013*

  This book chapter presents the multimodal architecture, the modules that should be present in any multimodal framework and usage scenario in the context of a living usability lab.

- *Nuno Almeida, António Teixeira. Enhanced interaction for the Elderly supported by the W3C Multimodal Architecture. Interacção 2013 – Conferência Interacção Pessoa-Máquina, Vila Real, 2013*

  This article describes the first implementation and architecture of the first version of the multimodal framework, it also describes the first prototype application, the news reader, which was used to test the framework at that current stage. The news reader application was used to

test future version of the multimodal framework and their modules.

In the context of the **modalities and modules of the multimodal framework**, several works were also presented in a number of articles. Speech is one the most important modalities created in the context of this work, since it is natural for humans to communicate through speech. The modality resulted in a versatile and generic modality that can easily be used in many applications without much effort.

- *Nuno Almeida, Samuel Silva, António Teixeira. Design and Development of Speech Interaction: A Methodology. Proceedings of HCI International 2014, June 2014*

  This paper presents a method based on the proposed multimodal framework, but emphasizing the implementation of interaction using speech. It also includes the description of the generic speech modality.

- *Nuno Almeida, António Teixeira, Ana Filipa Rosa, Daniela Braga, João Freitas, Miguel Sales Dias, Samuel Silva, Jairo Avelar, Christiano Chesi, Nuno Saldanha. Giving Voices to Multimodal Applications. Proc. HCI International 2015, Los Angeles, August 2015*

  Speech interaction is very important, either in speech recognition but also speech synthesis. The paper describes how different synthetic voices were recorded for European Portuguese and presents how elderly people react to each voice. The proposed generic speech modality is the entry point enabling the deployment of such desirable versatility.

Additionaly, other articles (Teixeira, Francisco, Almeida, Pereira, & Silva, 2014; Teixeira, Pereira, Francisco, & Almeida, 2015) present the services that provide support for the multilingual generic speech modality, describe how it can be configured and how services and modality work together.

Multi-device interaction was one key feature to solve in this work and the different approaches accomplishing this point were published in:

- *Nuno Almeida, Samuel Silva, António Teixeira, Diogo Vieira. Multi-Device Applications Using the Multimodal Architecture. In D. Dahl (Ed.), Multimodal Interaction with W3C Standards - Toward Natural User (1st ed., pp. 367–383). Springer International Publishing, 2017.*

  This book chapter, published in the first book focusing the community efforts aligned with the W3C standard approach to Multimodal Interaction, describes the evolution of the multimodal architecture and framework, mostly focused in the interaction manager (IM), and how the architecture supports multi-device scenarios.

Initial proposals of the use of the architecture in multi-device scenarios were also published in a W3C workshop related to the SCXML standard (Almeida, Silva, & Teixeira, 2014b) and in (Almeida, Silva, Santos, & Teixeira, 2016).

When speaking of multimodal interaction, **fusion** is an important topic, enabling multimodal events to be combined and fused into new ones giving them a new meaning. In this regard, we highlight:

- *Nuno Almeida, António Teixeira, Samuel Silva, João Freitas. Fusion of Speech with other modalities in a W3C based Multimodal Interaction Framework. Proc. IberSpeech, Lisboa, November 2016*

  The article describes how fusion was achieved and integrated in the proposed multimodal architecture. It presents a simple method to allow developers to configure and adopt fusion features in their applications.

The creation of new **applications** and the adoption of the multimodal architecture and framework allowed to iteratively improve the architecture. Several works were published presenting the developed applications and how they have adopted the framework.

- *António Teixeira, Nuno Almeida, Carlos Pereira, Miguel Oliveira e Silva, Diogo Vieira, Samuel Silva. Applications of the Multimodal Interaction Architecture in Ambient Assisted Living. In D. Dahl (Ed.), Multimodal Interaction with W3C Standards - Toward Natural User (1st ed., pp. 271–291). Springer International Publishing, 2017.*

  This book chapter presents the framework and describes some application that have fully adopted the multimodal framework developed in the context of this work.

- *António Teixeira, Flávio Ferreira, Nuno Almeida, Samuel Silva, Ana Filipa Rosa, José Casimiro Pereira, Diogo Vieira. Design and Development of Medication Assistant Elderly-centred Design to Go Beyond Simple Medication Reminders. Universal Access in the Information Society, June 2016 [IF 2014: 0.475 (Q4, 23/24)]*

This journal article presents a mobile application, a medication assistant, which adopted the main concepts of the architecture.

Other publications describing the applications that adopted the framework and providing and overall depiction of the evolution of the framework are: (Ferreira et al., 2013, 2014; Hämäläinen et al., 2015; Teixeira, Ferreira, et al., 2013; Teixeira, Hämäläinen, et al., 2013)

1.5 Structure

This thesis is divided in six chapters, organized as follows. This chapter introduced the motivation for this work, main challenges and objectives.

The second chapter presents the background and related work, covering Interaction in general, Multimodal Interaction, Modalities, Architectures, and multimodal interaction considering multiple devices.

Chapter 3 presents a proposal of a multimodal framework including the requirements to create such framework, an overview of the proposed architecture and a brief description of the modules required to create the framework.

In chapter 4, we present details regarding the implementation of the multimodal framework and the created modalities.

Chapter 5 describes each developed application in the iterative process that enabled evolving and making improvements to the multimodal framework. Those applications serve as proof of the validity of the proposed architecture and developed framework, at each stage, including a medication assistant, a personal life assistant, multi-device applications, and an application targeting children with special needs.

In the last chapter, we present the main conclusions for this work, with a summary and discussion of the main outcomes and future work.

Chapter 2

Background and Related Work

This chapter presents an overview on concepts and literature deemed relevant to assess how the current state-of-the-art provides support to designing and developing multimodal interaction in scenarios as those provided by AAL. In this context, the more relevant areas to present are human-computer-interaction, describing it in general, and then multimodal interaction with a brief overview regarding its main features and concepts. Considering that a wide range of technologies is available that support natural user interaction, which can be used in multimodal interaction, we deemed it relevant to present the ones that are currently more common. This background is described in sections **2.1**, **2.2** and **2.3**.

In the second part of the chapter, a survey of related work is presented including several works describing different approaches that enable the creation of multimodal interactive systems. It is divided into four sections **2.4, 2.5, 2.6** and **2.7.**

2.1 Human-Computer Interaction

In the literature we find several definitions for Human-Computer Interaction. Hewett et al. (1992) states that "Human computer interaction can be defined as the discipline concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them". Dix et al (1993) defines that "HCI is the study of people, computer technology and the ways these influence each other. We study HCI to determine how we can make this computer technology more usable by people". Other definition given by Carroll (2002) states that "HCI is the study and practice of usability. It is about understanding and creating software and other technology that people will want to use, will be able to use, and will find effective when used." A more simple definition is given by Tripathi (2011): "The human–computer interaction can be described as the point of communication between the human user and the computer. The flow of information between the human and computer is defined as the loop of interaction".

In the interaction between humans and computers, the output of the human is the input of the computer and vice-versa. Input and output in the computer is accomplished through the input/output devices and the range of devices that exist nowadays enables that many types of information can be entered and obtained from a computer. For humans, the input happens through the senses and the output through motor controls. For input, humans use mainly vision, hearing and touch and for output movement of finger/hand, voice, eyes, head, etc. (Dix, 2009)

Human-Computer interaction covers several aspects: methodologies and processes for designing interfaces, methods for implementing interfaces, techniques for evaluating and comparing interfaces, developing new interfaces and interaction techniques, and developing descriptive and predictive models and theories of interaction (Tripathi, 2011).

Improving the interaction between users and computers is the primary goal of Human-Computer Interaction, creating simpler interaction systems that are more usable, cancelling the negative aspects of the user experience, such as frustration and annoyance, and improving the positive aspects, enjoyment and engagement, while using the system. (Preece, Paine, & Rogers, 2015)

Ever since the beginning of the personal computing era, more than three decades ago, the main input system that users had for interacting with these machines was a physical keyboard. Shortly after, the introduction of the mouse as a pointer device revolutionized personal computers making them easier and faster to use.

With the development of the mouse, WIMP interfaces (windows, icons, menus, pointer) became feasible, introducing a style of interaction that uses these elements in the graphical user interface. Indeed, WIMP interfaces use graphics as a key element (Hinckley & Wigdor, 2002).

WIMPs (Figure 2-1) are interfaces where a window will run a self-contained program, isolated within that window from other programs running at the same time (used to create multi-program operating systems). The icons act as shortcuts to actions to be performed by the computer (such as execute a program). Menus are text-based or icon-based selection systems to again select and execute programs or sub-programs. Finally, the pointer is an onscreen symbol that represents the movement of a physical device to allow the user to select elements on an output device such as a monitor (Hinckley & Wigdor, 2002).

Figure 2-1 - Representation of the WIMP interfaces, from left to right, the four basic concepts: Windows, Icons, Menus and Pointer

The primary benefit of this style of interface over the predecessor keyboard-only and text-only (command line) interfaces is to improve the human-computer

interaction by enabling better ease of use for non-technical people, both novice and power users. Know-how can be ported from one application to the next, given the high consistency between interfaces.

This kind of interface allowed the growth of personal computers, being the primary way of interaction between humans and technology nowadays.

2.2 Multimodal Interaction

The definition of multimodal interaction (MMI) is directly related with human-computer interaction. Bernsen (2008) states "A multimodal interactive system is a system which uses at least two different modalities for input and/or output.". Jaimes and Sebe (2007) present a similar definition, that a multimodal system "is simply one that responds to inputs in more than one modality or communication channel (e.g. speech, gesture, writing and others)".

Other definitions are presented in literature and define multimodal systems as something more complex than the previous definitions. Oviatt (2003) states that multimodal interfaces are accomplished by processing multiple (two or more) modalities for input combined (speech, pen, touch, manual gesture, gaze, and head and body movements) and coordinated with an output system. The definition is related with natural interfaces, recognizing forms of the user language and behavior. Bourguet (2003) presents a similar definition: "Multimodal interaction refers to interaction with the virtual and physical environment through natural modes of communication such as speech, body gestures, handwriting, graphics or gaze". Those definitions are more related with natural language inputs.

Multimodal interaction systems offer multimodal capabilities for the exchange of information between humans and computers. The information from different communication channels has to be interpreted by the system (Dumas, Lalanne, & Oviatt, 2009). Such system must provide to users a set of modalities enabling the interaction with the system.

Reeves, Lai, and Larson (2004) presents two goals for the multimodal interaction: first, obtain a natural interaction, one that is similar to the communication between humans; second, increase the user experience by having more robust interaction and by using redundant or complementary information.

Some multimodal systems are already available for public use, for instance mTALK (Johnston, Fabbrizio, & Urbanek, 2011) is a multimodal browser which enables the use of mobile applications endowing them with natural modalities such as speech, touch and gestures. There are other more used and know, but their solutions are mainly proprietary and closed. An example of the ford SYNC (Ghangurde, 2010), which is a system for entertainment control and communication controlled by voice. Other examples are the ones that we increasingly use in our phones and computers, as they came installed by default, Google Now[2], Siri[3] and Cortana[4], Figure 2-2 shows the logos of each enumerated personal assistant.

Figure 2-2 - Logos of the personal assistants available for different platforms. From left to right Google Now, Siri and Cortana

2.2.1 Modalities

Two main and quite different definitions for "modality" are found in literature, first the Bernsen (2008) defines: "a modality of information representation is a way of representing information in some physical medium.

[2] https://www.google.com/search/about/learn-more/now/
[3] http://www.apple.com/ios/siri/
[4] https://www.microsoft.com/en/mobile/experiences/cortana/

Thus, a modality is defined by its physical medium and its particular 'way' of representation". The other definition is presented by the standard W3C (Michael Bodell et al., 2012), and states that a modality is a component responsible to manage the input/output modalities on the device. In the W3C modalities can assume a more complex form, a modality is allowed to have functionalities that logically would belong in separated modalities, for instance a graphical interface with touch capabilities.

Modalities can be classified in different ways and different authors may present slightly different approaches. Table 2-1 shows the classification of modalities presented by Karray et al. (2008) that encompasses all modalities into three main categories: visual, audio and sensor based.

Table 2-1- Modalities classification from (Karray et al., 2008)

1. Visual-Based	Facial Expression Analysis
	Body Movement Tracking (Large-scale)
	Gesture Recognition
	Gaze Detection (Eyes Movement Tracking)
2. Audio-Based	Speech Recognition
	Speaker Recognition
	Auditory Emotion Analysis
	Human-Made Noise/Sign Detections (Gasp, Sigh, Laugh, Cry, etc.)
	Musical Interaction
3. Sensor-Based	Pen-Based Interaction
	Mouse & Keyboard
	Joysticks
	Motion Tracking Sensors and Digitizers
	Haptic Sensors
	Pressure Sensors
	Taste/Smell Sensors

Bernsen (2008) classifies what is a modality in a different, more granular way. He starts by classifying modalities as Linguistic, Analogue, Arbitrary, and Structure, in the first level, and then addresses their expansion in generic, atomic and sub-atomic levels. Table 2-2 presents the complete classification of modalities by Bernsen. Just by observing the number of classifications for

modalities, it is notable the number of possible modalities that can be developed and integrated in multimodal systems.

Touch is already widely used, they are very common in our smartphones that we use every day. Other modalities not so used but also very import are the body gestures, using the Kinect or video processing and eye-gaze.

Some modalities are more common because the technology is more evolved and they are more natural, speech input and speech output are now very frequent modalities given that they are very natural for humans. They can be used to give simple commands and receive simple messages, or the can be combined with Natural Language Understanding (NLU) and Natural Language Generator (NLG) to interact through a spoken conversation, which for humans is the primary way of communication.

Table 2-2 - Modalities classification from (Bernsen & Dybkjær, 2010)

Super level	Generic level	Atomic level	Sub-atomic level
Linguistic modalities >>	1. Static analogue graphic elements		
	2. Analogue acoustic elements		
	3. Analogue haptic elements		
	4. Dyn. Analogue graphic elements	4a. Sign language discourse	
		4b. Sign language lab.-keyw.	
		4c. Sign language notation	
	5. Static graphic language	5a. Static text	5a1. Typed text
			5a2. Hand-written text
		5b. Static labels/keywords	5b1. Typed lab.-keyw.
			5b2. Hand-written lab.-keyw.
		5c. Static notation	5c1. Typed notation
			5c2. Hand-written notation
	6. Acoustic language	6a. Spoken discourse	
		6b. Spoken labels/keywords	
		6c. Spoken notation	
	7. Haptic language	7a. Text	

Super level	Generic level	Atomic level	Sub-atomic level
		7b. Labels/keywords	
		7c. Notation	
	8. Dynamic graphic language	8a. Dynamic text	
		8b. Dynamic labels/keywords	
		8c. Dynamic notation	
		8d. Visual spoken discourse	
		8e. Visual spoken labels/keywords	
		8f. Visual spoken notation	
Analogue modalities >>	9. Analogue static graphics	9a. Images	
		9b. Maps	
		9c. Compositional diagrams	
		9d. Graphs	
		9e. Conceptual diagrams	
	10. Analogue acoustics	10a. Images	10a1. Gestures
			10a2. Expression
		10b. Maps	
		10c. Compositional diagrams	
		10d. Graphs	
		10e. Conceptual diagrams	
	11. Analogue haptics	11a. Images	11a1. Gestures
			11a2. Expression
			11a3. Body action
		11b. Maps	
		11c. Compositional diagrams	
		11d. Graphs	
		11e. Conceptual diagrams	
	12. Analogue dynamic graphics	12a. Images	12a1. Gestures
			12a2. Expression
			12a3. Body action
		12b. Maps	
		12c. Compositional diagrams	
		12d. Graphs	
		12e. Conceptual diagrams	
Arbitrary modalities >>	13. Arbitrary static graphics		
	14. Arbitrary acoustics		
	15. Arbitrary haptics		
	16. Arbitrary dynamic graphics		
Structure modalities >>	17. Static structure graphics		
	18. Acoustic structure		
	19. Haptics structure		

Super level	Generic level	Atomic level	Sub-atomic level
	20. Dynamic graphic structure		

2.2.2 Fusion and Fission of Modalities

With the availability of many different interaction modalities, some issues arise that need to be solved. While interacting with a multimodal application, users can interact with the system using multiple modalities simultaneously or in sequence. For instance, users can make the same command using different modalities or commands that complement each other. This variety of possibilities needs to be managed.

Events from multiple input modalities can be extracted, recognized and then fused into other event (Dumas, Ingold, & Lalanne, 2009). The main goal of fusion of modalities is to extract meaning of a set of events coming from the input modalities, fusing the information of one or more events into one with the completed information. An example of is the "put that there" (Bolt & Bolt, 1980), where an event is generated by a speech modality and two sequential touches in places of the screen, the first touch is the object (that) and the second the place (there).

The fusion engines can be classified into three levels according to the type of data generated by modalities, feature level or early fusion, decision level or late fusion, and hybrid (Atrey, Hossain, El Saddik, & Kankanhalli, 2010). In the first the fusion engine processes the features extracted by the input modalities and operates at low level. The second operates at a semantic level at high level. The third is a hybrid of the first two.

Events can be combined in different ways. Two well-known models conceptualizing those combinations are CASE (Concurrent, Alternate, Synergistic, Exclusive), presented in (Nigay & Coutaz, 1993); and CARE (Complementarity, Assignment, Redundancy and Equivalence) described in

(Coutaz et al., 1995). The first focus on the fusion at the fusion engine level, and the second in the interaction level between user and machine.

The fission concept is the dual of the fusion of modalities, one receives events from the inputs the other is responsible to select the output modalities. Figure 2-3 shows the concept of a multimodal system, input modalities sends the events to the fusion, the fused events are then sent to the fission, which selects the output modalities to present the information.

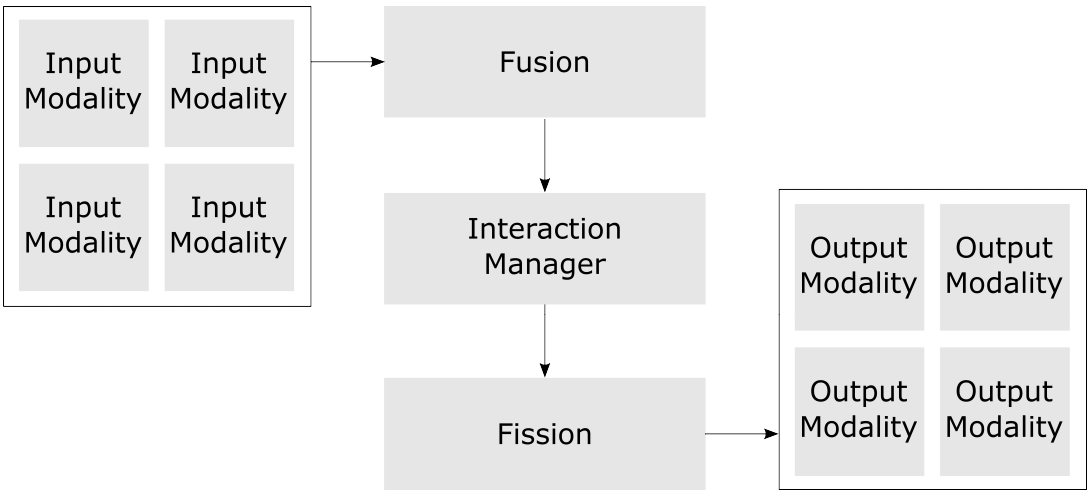

Figure 2-3 - Multimodal Concept, Fusion of the input modalities and fission to output modalities

2.3 Supporting technologies for modalities

New technologies that enable interaction have emerged and evolved in recent years, technologies that supports more natural and richer interaction between human and machine. Many of those technologies are widely available and developers can use them into their application.

2.3.1 Speech Recognition (ASR)

The preferred way for humans to interact is spoken conversation, so the main natural ways of interaction is speech (Tadeusiewicz, 2010). Performances in automatic speech recognition (ASR) are constantly improving and getting

better results, this contributes for user acceptance and to be included in multimodal systems, which is the case of mTalk (Johnston et al., 2011).

ASR systems coverts the acoustic speech signal into a sequence of words, without attempting to attribute any sort meaning to the obtained sequence. The most common approach is statistical formulation of speech recognition problem. This formulation (Jurafsky & Martin, 2000; C. Martins, 2008) seek to find the most probable word (W), given an acoustic observation (O) by computing the probability P(W | O) for all word sequence and choosing the one with highest probability:

$$\widehat{W} = argmax_W \, P(W|O)$$

In general, a variety of assumptions and simplification are made to calculate P(W | O): first, words are typically decomposed in phonetic units (U), representing the sounds used to distinguish different words. Applying Bayes' theorem, the problem is reduced to finding:

$$\widehat{W} = argmax_W \, P(W|O) = argmax_{W,U} = P(O|U)P(U|W)P(W)$$

The first part, P(O | U) is commonly designated as the acoustic model; P(U | W) is the lexical pronunciation model and P(W) is known as language model.

The acoustic model is a statistical model with the relation between audio signal and phonemes. The model can be trained from audio recordings with the aligned transcript. Most common acoustic models are based on Hidden Markov Models (HMM) (Jurafsky & Martin, 2000). These models are language dependent and for each language that we want to do speech recognition new models must be trained.

Two main types of language models are common: grammar based and statistical language models. Grammars are the simplest and are used when the number of sentences to be recognized are reduced.

```
$digit = one | two | three | four | five | six | seven | eight |
    nine | zero | oh ;
```

```
$digits = [*sil%% | *any%%] *$digit [*sil%% | *any%%];
```

Code 2-1 – Sample grammar for digit strings (from (Cassidy, n.d.)*)*

These statically language models contain the probability of sequence of words, typically represented by an n-gram. This models stores the probability of each word given the preceding sequence. Grammars limits the amount of sentences, but is more accurate when recognizing sentences.

Compared to other modalities, speech interaction does not need the full attention of its user, the user can interact with one system and still perform other tasks. The user doesn't have to be near the installed system, but only in range, so he/she can hear the speech synthesis and the microphone listen to him. Users can benefit of speech interaction if they are in a quiet place, noisy environments affect the performance of the speech engines. If there is a big noise in the environment, the current speech engines will not be able to distinguish that from the speech of the users. It is one of the main weakness of current speech recognition engines, it also applies to humans in noisy environments.

2.3.2 Speech synthesis

Speech synthesis complements speech recognition technologies on a spoken conversation. Voice synthesis consists of the production of voice by a machine using algorithms, rules and acoustic parameters.

In general, it has two main modules: text processing and signal generation as presented in Figure 2-4.

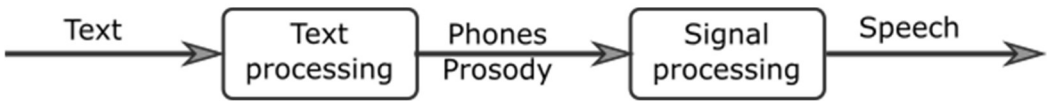

Figure 2-4 – Main modules of a speech synthesis system, providing a text the first module processes the text resulting in the phones and prosody. This result is processed in the signal processing module to create the speech.

The text processing module does some task, it starts by analyzing the text to detect the sentence punctuation, normalize symbols. Converts the text to phonemes and generates prosody.

Signal generation can be done using several techniques such as diphone concatenation, unit selection, formants, or more recently, HMM-based synthesis (HTS) (Taylor, 2009).

For concatenative methods and even HTS the data base with samples or model parameters is known as a voice and it is created based on human recordings. An example of recent work for Portuguese was the creation of four new Portuguese voices (Almeida et al., 2015), male and female, for young and older adults.

Performance and the quality of the generated speech are improving, synthetic voices are more natural. This fact also contributes for a better user acceptance (Johnston et al., 2011). Speech synthesis engines use a model for each voice, each model is created with the voice features, male or female, young or adult. Also it is possible to configure the rate and pitch in the speech synthesis engine. Having many voices for each language is important, for instance, since different users have different preferences. Furthermore, preferences depend on the user context. The creation of new voices represents a great effort, since each synthesized voice is created with the recordings from a single person.

2.3.3 NLU & NLG

Language understanding technologies (NLU) and Natural Language Generation (NLG) combined with a Dialog Manager (Revuelta-Martínez, Rodríguez, García-Varea, & Montero, 2013), aim to provide users with a natural feel of conversation. Using NLU text or speech can be interpreted to generate semantics that the computer can understand, semantic information can be translated to commands. NLG can generate natural language (J. C. Pereira & Teixeira, 2015; Reiter & Dale, 2006; Teixeira, Pereira, et al., 2015), which the user is familiarized, to deliver information to the user.

2.3.4 Multi-touch

Multi-touch interfaces use a touch sensing surface, recognizing the presence of one or more point of contact. Nowadays, multi-touch is associated with screens, allowing the user to interact directly with the information displayed by the screen and so it is considered one of the most natural ways of interaction (Holzinger, 2003). Since the widespread of smartphones and tablets, the technology is widely available. The multi-touch allows users, for example, to select objects, drag-and-drop, rotate, and make zoom.

2.3.5 Gestures

Use of gestures in interfaces appeared to the public and gained popularity, is becoming more and more common (Saffer, 2008). It needs recognition of the body gestures that the user performed. There are two main techniques to recognize such gestures: one based in motion sensors, such as accelerometer or gyroscope, other based on video and image processing. Two known devices, one from each technology, are the Nintendo Wii Remote (Saffer, 2008) and Microsoft Kinect (Tashev, 2013). In the first the user needs to carry the device and the gesture recognition is done by detecting the motion of the remote, in the Kinect detects the user and its movements. In this case the user must be placed in front of the cameras. More information on the how to use Kinect can be found in (Microsoft Corporation, 2013) or (Jana, 2012).

2.3.6 Eye tracking

Eye tracking is a technology that allows tracking the direction of the user eyes ("Tobii: this is eye tracking," 2015) and, from there, after a calibration step, infer the location to where the user is looking. It consists on a camera focusing the user's eyes, and modern techniques use infrared light and near-infrared light to create a corneal reflection. Although it is mostly used by people with disabilities, which cannot use traditional ways to interact, it can be combined,

for instance, with speech to improve the confidence of the recognition (Vieira et al., 2015).

2.3.7 Other modalities

Input modalities are given most relevance, but output modalities are also important, since they delivery information to the user. Several output modalities can be considered such as display imagery, avatar, graphic outputs, sound output. Each can deliver a message to the user in a different way or they can be combined to present the information.

2.4 Multimodal Architectures

In literature, several multimodal architectures are presented. They can be separated into categories, we have identified some categories that each multimodal architectures fit. The following subsection identifies the architectures and presents example of each architecture type.

2.4.1 Agents based architecture

Bellifemine et al. (2003) states that "Agent-based systems are intrinsically peer-to-peer: each agent is a peer that potentially needs to initiate a communication with any other agent as well as it is capable of providing capabilities to the rest of the agents. The role of the communication is very important in an agent-based system, and its model is based on three main features: 1. agents are active entities […] 2. agents perform actions and communication is just a type of action. […] 3. communication carries a semantics meaning".

In this kind of architectures each module is implemented as an agent, which exchange messages between them. Usually it uses a facilitator or mediator between agents, that is also an agent. Examples of agent based architectures are QuickSet (Cohen et al., 1997a), HephaisTK (Dumas, Lalanne, & Ingold,

2009). Other architectures based on agents can be decentralized and agents communicates with the agents they need, one example of this architecture is AdaptO (Teixeira et al., 2011). In Figure 2-5 and Figure 2-6 presents examples of the architectures with centralized and decentralized, respectively.

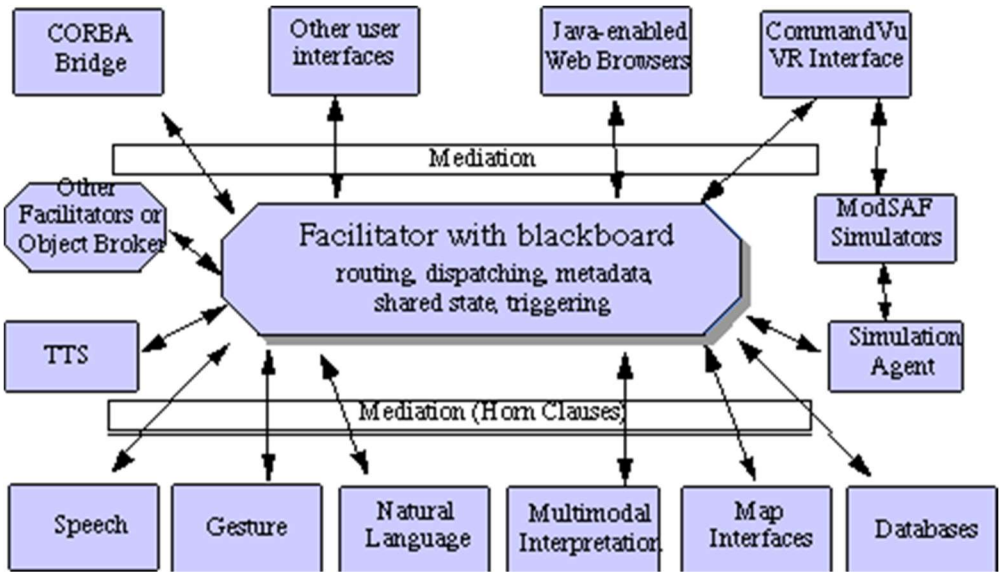

Figure 2-5 – Agent based architecture, QuickSet. From (Cohen et al. 1997a)

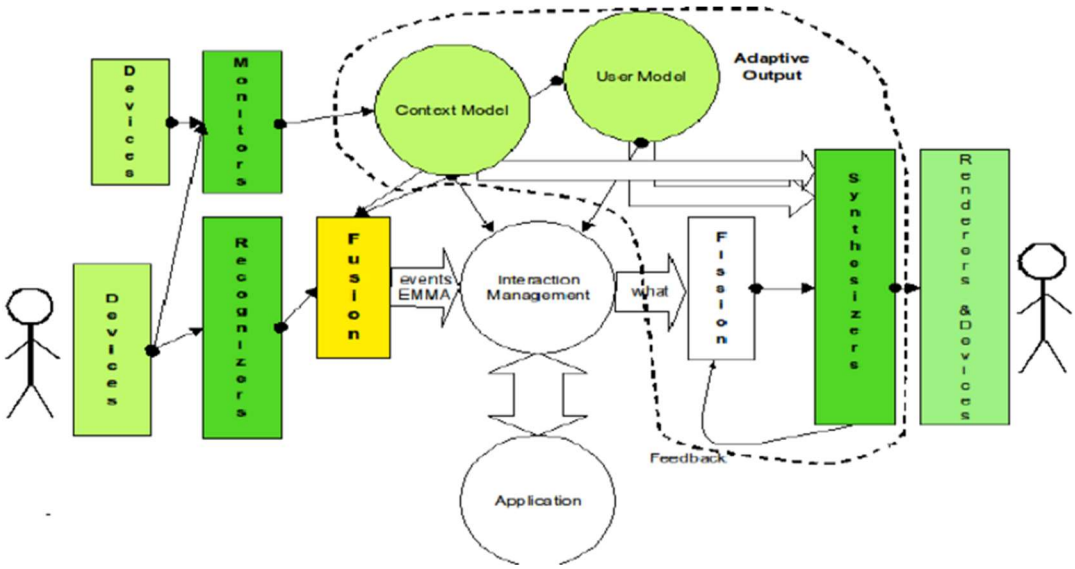

Figure 2-6 – Agent based architecture, AdaptO. From (Teixeira et al., 2011)

2.4.2 Components based architectures

These architectures are defined by having a core or a kernel, which loads statically or dynamically the components of the system, Figure 2-7 presents an overview of architectures based on components. Despite these architectures allow the dynamically loading of the components, it does not allow their usage in distributed scenarios. OpenInterface (Serrano et al., 2008) and ICARE (Bouchet & Nigay, 2004) are examples of components based architectures.

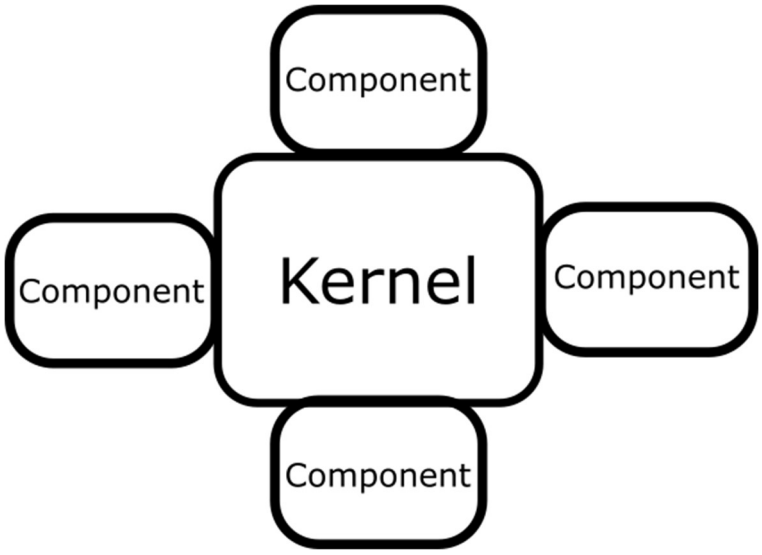

Figure 2-7 – Overview of a component based architecture

2.4.3 Layer based architectures

In a layer based architecture modules of the architecture are classified according to their features. The modules of each layer communicate with the modules of the next and/or previous layer, typical layers are associated with input, multimodal core or integration and output. Figure 2-8 presents an example of a layer based architecture, with three distinct levels. Examples of the architectures are the one presented in the Figure 2-8 (Mcglaun, Mcglaun, Lang, & Rigoll, 2004) and the MUDRA (Hoste, Dumas, & Signer, 2011).

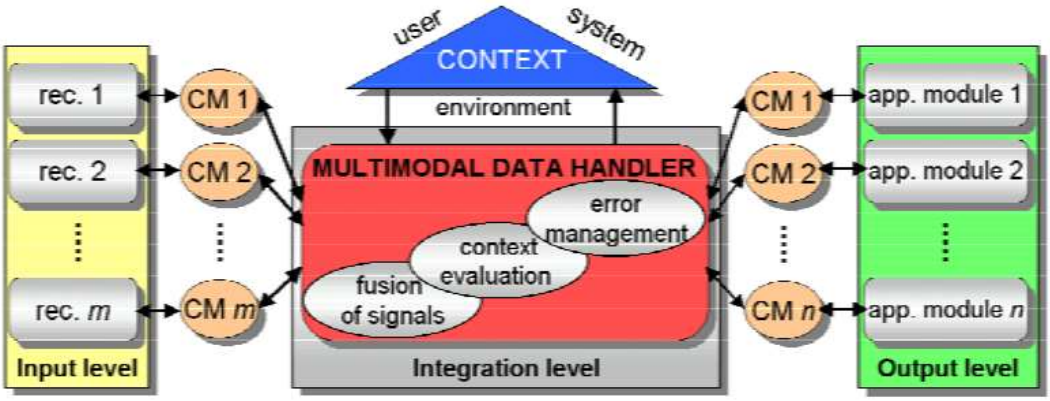

Figure 2-8 - Layer based architecture, from (Mcglaun et al., 2004)

2.4.4 Server based architectures

These architectures are more commonly used in mobile scenarios, since mobile devices have more limited execution performance, but it can be used for other scenarios. In these architectures the devices only run a client that interact with a server, the server contains the recognizers and the multimodal core, Figure 2-9 present an architecture based on a server. The work SmartWeb Handheld (Sonntag et al., 2007) and the architecture proposed by Niklfeld, Finan, and Pucher (2001) are examples of this type of architecture.

2.4.5 Other distributed architectures

Besides the distributed architecture based on agents there are other distributed architectures that are quite relevant.

The I-TOUCH (Pocheville, Kheddar, & Yokoi, 2004) is one example, other example is the W3C Multimodal Interaction Architecture (Bodell, Dahl, Kliche, Larson, & Porter, 2012; Dahl, 2013). The W3C architecture is particularly important since it is a recognized standard for multimodal interaction. A more detailed presentation of the architecture follows.

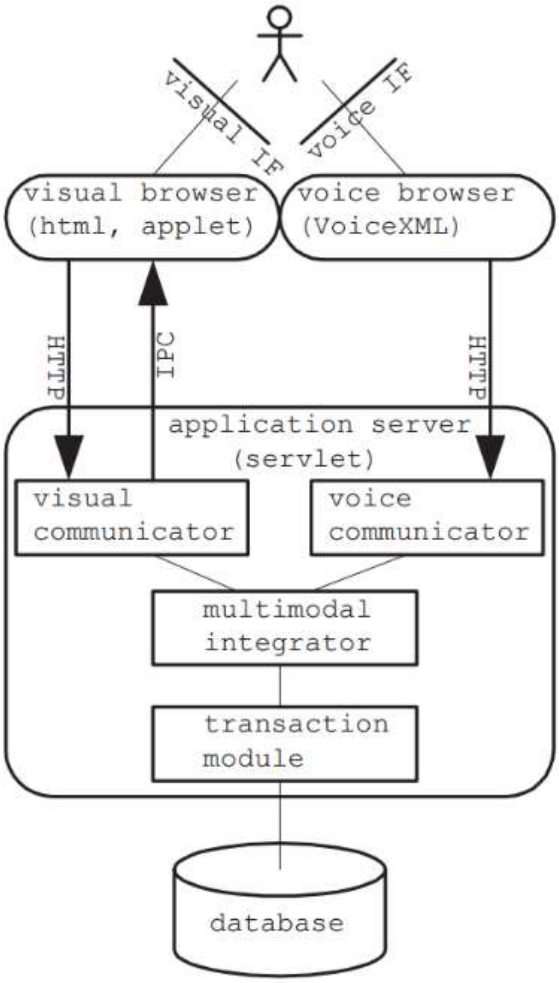

Figure 2-9 – Example of a server based architecture, From (Niklfeld et al., 2001)

2.4.5.1 The W3C Standards for a Multimodal Interaction Architecture

Since many different multimodal frameworks are available with different characteristics, the W3C created the Multimodal Architecture to provide a standard to help technical development and innovation.

The W3C multimodal architecture (Bodell et al., 2012; Dahl, 2013) is a standard, which defines several aspects of multimodal systems, from the components to languages used for the communication among components.

The main components of the standard are the runtime-framework, interaction manager, data component and modalities as presented in Figure 2-10. Modalities cannot communicate directly with the application and must use

the event transport layer provided by the runtime-framework to communicate with the interaction manager. Events exchanged by interaction manager and modalities are defined as MMI life cycle events, which can encapsulate EMMA messages that carries the information of events.

2.4.5.1.1 Interaction Manager

The MMI architecture recommends the use of a state machine, which can be defined in SCXML. It is responsible to receive and respond to all life cycle events from modalities. Also, based on the states of the SCXML the interaction manager can generate new life cycle events to send to modalities. The SCXML

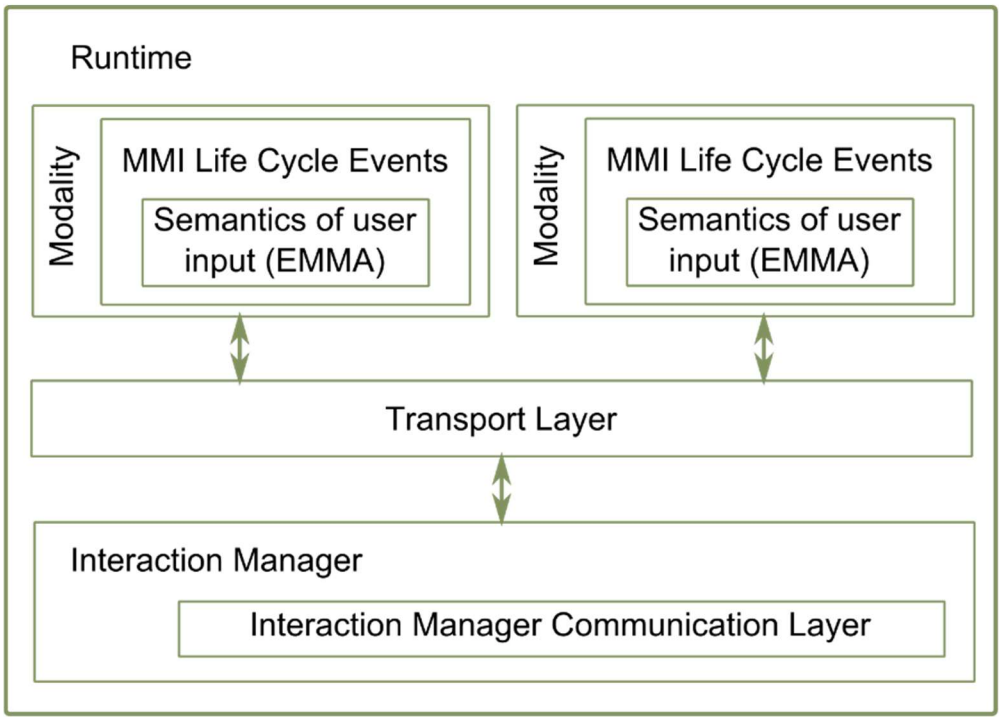

Figure 2-10 – Components of the Multimodal Architecture (runtime, interaction manager, data model and modalities)

(Barnett et al., 2015) is a markup language that defines a state chart machine and the data model. Its objective is to provide the application logics to the existing framework.

The basic concepts of a state machine are <state>, <transition> and events (SCXML Events). One state machine contains a data model <datamodel>with a set of <data>, and a set of states, each state contains a set of transitions that define how the state machine reacts to the incoming events from modalities, the state machine can also generate events. When running, a state machine has a single active state. When an event occurs, the machine tries to matches the event to the transitions on the active state. If a transition matches, then the target state of that transition is set as the new active state.

In SCXML, there are some extensions to a basic state machine. State machines can have executable content via conditions like <if>, <elseif>, <else>, <foreach>; executable scripts <script>, <raise> to raise events in the current SCXML, <send> to send messages to external entities or modalities, <assign> to modify the data model. It also has two elements to execute content upon entering <onentry>or exiting <onexit>a state.

Extending the basic structure, states can also have sub or parallel states. All states inside an active parallel state are considerable active. However, when a transition occurs to an outside state, they all become 'non active'.

2.4.5.1.2 Event Description and Communication

EMMA (Extensible MultiModal Annotation markup language) is one of the standards adopted by the MMI architecture, this language is used to describe the events that are generated by the different input modalities (Baggia et al., 2009). Those generated messages from input modalities, such as speech recognition, natural language text, touch, etc… are encapsulated within a Lifecycle event and sent to the interaction manager. The EMMA messages carries three types of data: (1) instance data, the application-specific markup it is the input information more important for the EMMA consumer: (2) data model, the constraints on structure and content of an instance. Normally the data model is implicit. (3) metadata, the annotation associated with the data

that is being sent. Code 2-2 show an example of an EMMA message generated by a touch modality.

```
<emma:emma emma:version="1.0">
    <emma:interpretation
       id="touch1"
       emma:confidence="1.0"
       emma:start="1319042815785"
       emma:medium="touch"
       emma:mode="object">
         <object>pause_button</object>
    </emma:interpretation>
 </emma:emma>
```

Code 2-2- Example of an EMMA message

The fusion of events is a topic addressed by the W3C multimodal architecture (Michael Bodell et al., 2012), but it is not clear where a possible fusion module fits in the architecture. Recent work by (Schnelle-Walka, Radomski, & Mühlhäuser, 2014) propose a multimodal system based on the W3C multimodal architecture, where both fusion and fission are a part of the system. The multimodal system has a root interaction manager that communicates with fusion and fission, which can be seen as modality components to the root interaction manager and as interaction managers to other modality components. Figure 2-11 shows the proposed solution to include the fusion and fission module.

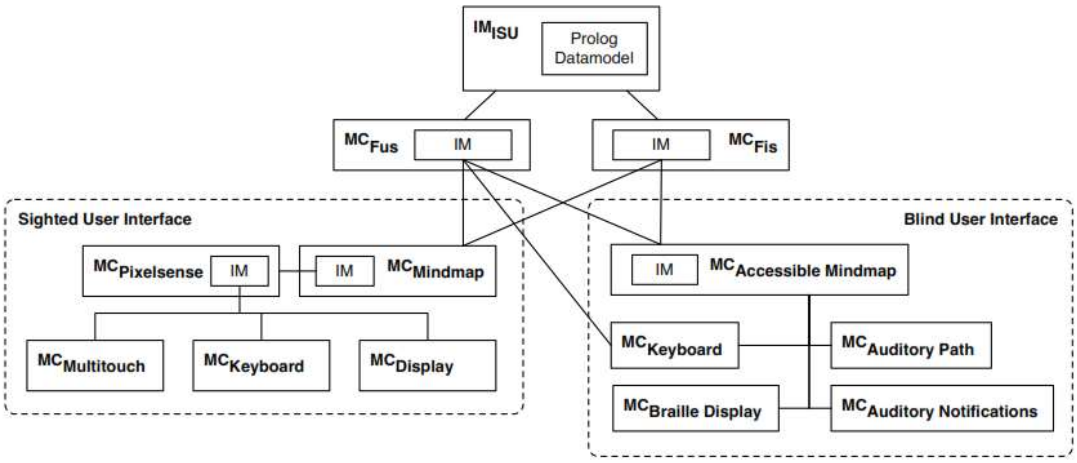

Figure 2-11 - Proposed multimodal architecture with fusion and fission modules. From (Schnelle-Walka et al., 2014)

2.5 Development tools & Frameworks

Over the years several frameworks have been proposed to enable and ease the creation of multimodal application. Literature related to multimodal interfaces shows that a number of frameworks are created with specific architectures and other based on the standard architecture proposed by the W3C (Cutugno, Leano, Rinaldi, & Mignini, 2012; Pous & Ceccaroni, 2010).

The following subsection presents development frameworks and tools that allow developers to simulate the interaction in mockups or dummy applications, using a set of different modalities. The frameworks provide support for the development of multimodal interaction applications.

2.5.1 QuickSet

QuickSet (Cohen et al., 1997b) is one of the first tools to appear, it is a system running on a hand held PC and desktop computer, aiming for a pen gestures and voice interaction on multiple devices. The architecture of this system is based on agents, which communicate through a wireless network. The modalities of the systems are agents and integrated using the Open Agent Architecture. An agent called "Multimodal integration agent" accepts the

information from the input agents and produces a unified multimodal interpretation. Figure 2-12 presents an overview of the QuickSet architecture based on agents.

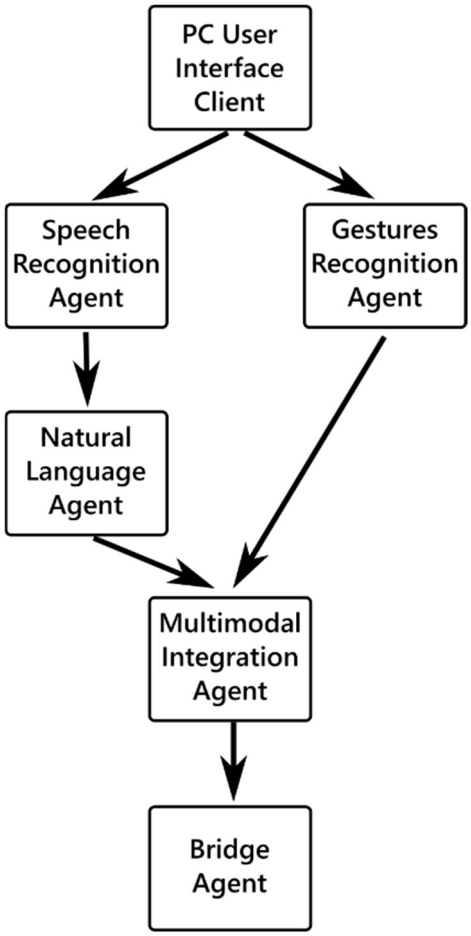

Figure 2-12 – Simple overview of the QuickSet architecture. From (Johnston et al., 1997)

2.5.2 ICON

ICON (Dragicevic & Fekete, 2001) is a visual editor to configure input devices (Figure 2-13) and connect inputs to an interactive application. It enables developers to view and edit relations between inputs and application, based on a dataflow model where modules can be linked between inputs and outputs.

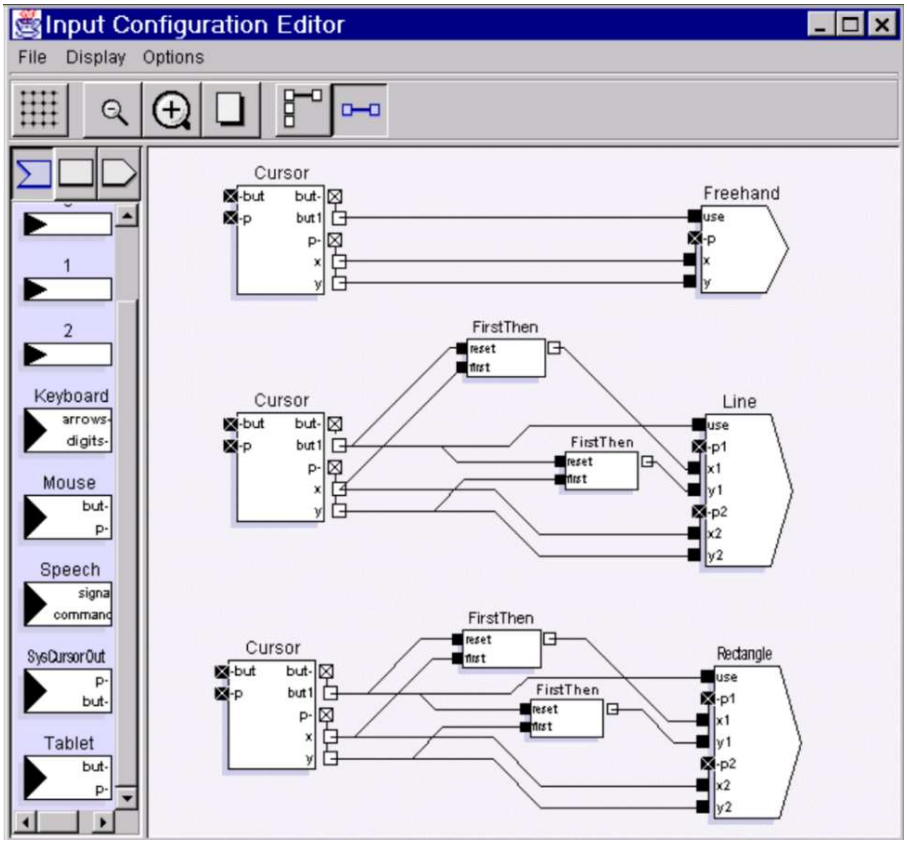

Figure 2-13 – Example of an input configuration, edited in ICON, from (Dragicevic & Fekete, 2001)

2.5.3 CrossWeaver

CrossWeaver (Sinha & Landay, 2003) allows designers to plan a multimodal application through a graphical user interface. Users draw user interfaces as storyboards, this sketches enables the exploration of multimodal scenarios, by executing the Storyboards to test the interactions. The prototyping tool supports a limited number of modalities, and is not suitable to integrate additional ones.

2.5.4 ICARE

ICARE (Bouchet & Nigay, 2004) (Interaction CARE), it is a generic platform, based on components to build multimodal applications. It targets designers instead of developers, allowing them to select modalities and detailed

combination, in a way to fulfill de CARE properties. The platform is not easily extensible and produces non-reusable components.

2.5.5 Openinterface

OpenInterface (Serrano et al., 2008) is a tool that allows for rapid development of multimodal interfaces. It includes two parts, the OpenInterface Kernel, a generic runtime platform and Sketch Multimodal Interactions (SKEMMI) a visual design platform. The first allows the integration of components, those are connected through a pipeline, which enables the connection and configuration of components, control of the life-cycle and the execution site. Figure 2-14 presents an overview of the OpenInterface Architecture.

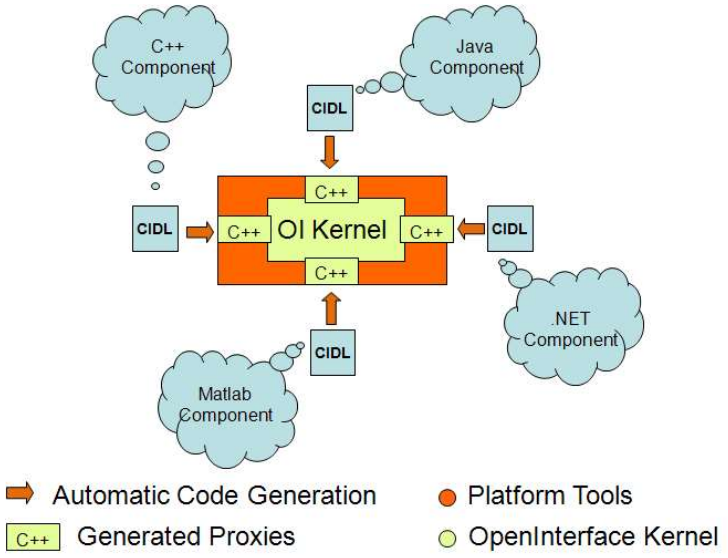

Figure 2-14 - Overview of OpenInterface Architecture (Lawson, Al-Akkad, Vanderdonckt, & Macq, 2009)

The SKEMMI allows to build the multimodal pipelines with minimal efforts, it is a design front-end, which supports multi-level of interaction design. Using design-by-demonstration or direct manipulation allows the modification and

composition of the application. Figure 2-15 shows the visual design of a simple multimodal application.

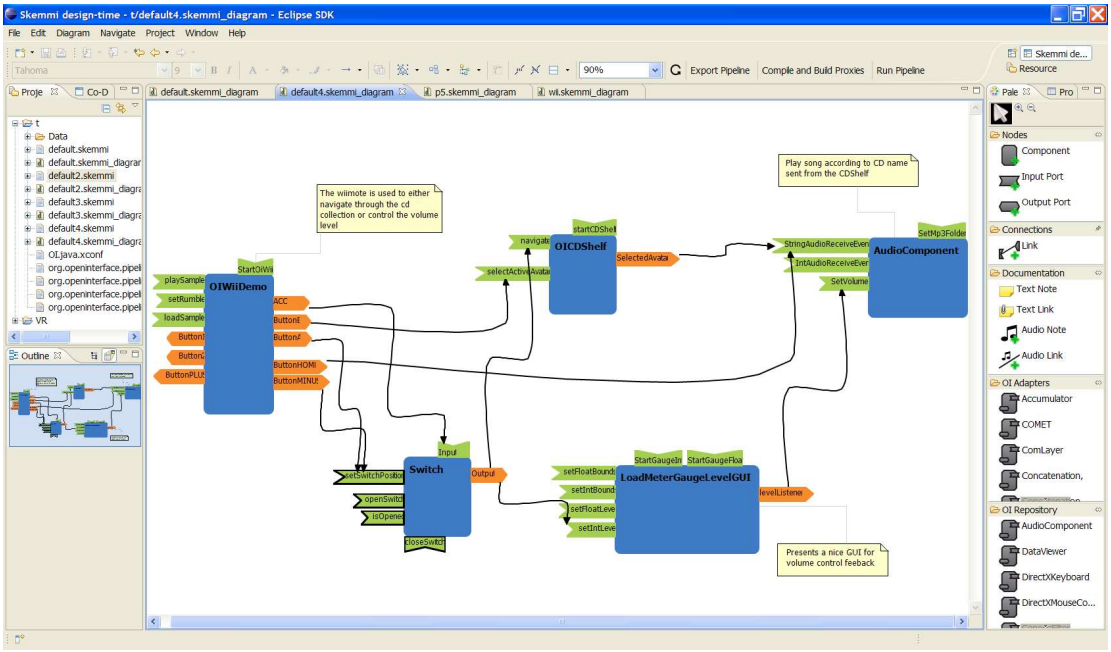

Figure 2-15 - SKEMMI plugin (Lawson et al., 2009)

2.5.6 HephaisTK

The HephaisTK (Dumas, Lalanne, & Ingold, 2009) is a toolkit that enables rapid prototyping of multimodal interaction. It offers developers the tools to create multimodal interfaces, has already a set of recognizers, and allows developers to create new ones. New modalities must comply with a set of rules for the communication between the modality and the toolkit. The communications are done with standard messages, W3C EMMA, to exchange events. The toolkit can manage fusion of modalities and user-machine dialog with a finite state machine, which is defined using the SMUIML markup language (Dumas, Lalanne, & Ingold, 2008). The language defines three layers (1) dialog, (2) events and (3) input/output as shown in Figure 2-16, and it is capable of handling the CARE properties.

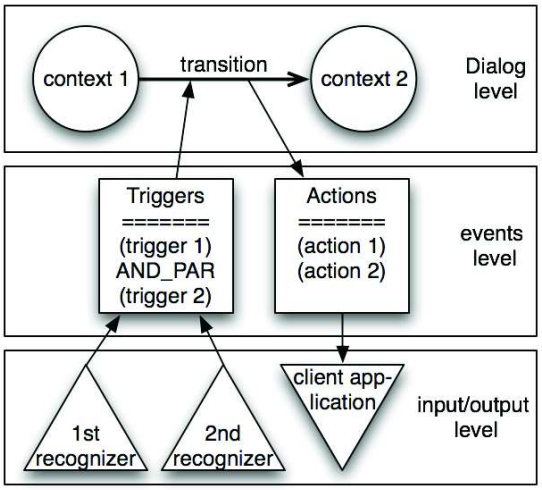

Figure 2-16 - SMUIML Architecture (Dumas, Lalanne, & Ingold, 2008)

One of the main components of HephaisTK is called "postman". It receives the messages from each recognizer stores them and is responsible to deliver the messages to agents that subscribed a specific type of messages. Figure 2-17 shows an overview of the HephaisTK architecture.

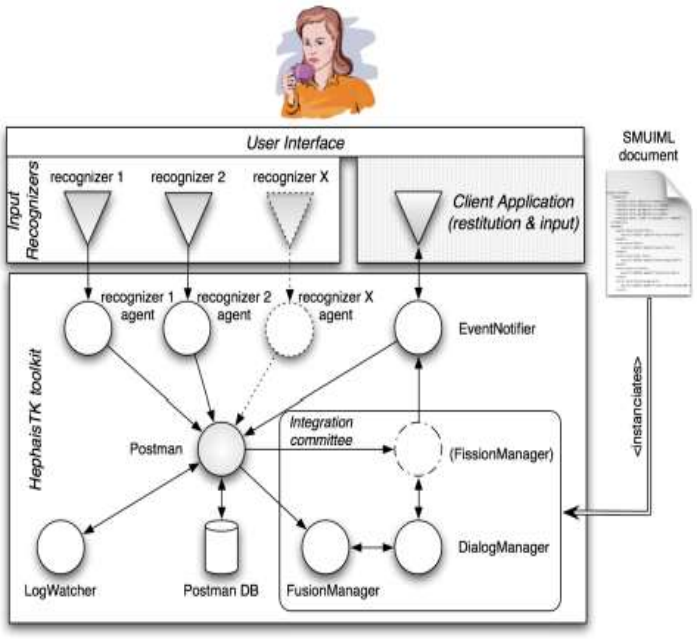

Figure 2-17 - HephaisTK Architecture.(Dumas, Lalanne, & Ingold, 2009)

2.5.7 Mudra

Mudra (Hoste et al., 2011) is a multimodal interaction framework that unifies low-level data and high-level semantics. Most oriented to multimodal fusion, aims to extract and fuse information to infer semantic interpretation. The architecture centers on a Fact Base (a fact is defined by a type and a list of atributes) in combination with a declarative rule-based language. Figure 2-18 presents the architecture of the framework.

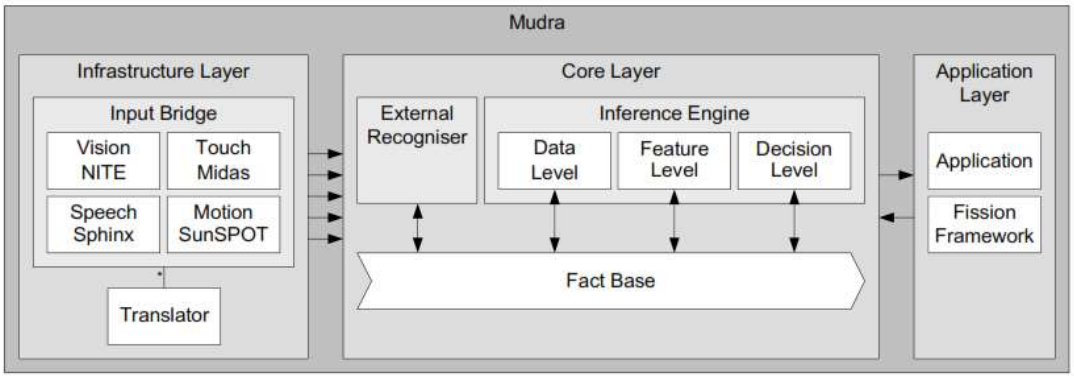

Figure 2-18 - Mudra Architecture

2.5.8 Manitou

Manitou (Hak, Dolezal, & Zeman, 2012) is a web platform, which enables developers to create multimodal applications with speech, gestures and other modalities. It includes a set of libraries and tools and a multimodal framework with those tools.

2.5.9 M3I: Mobile Multimodal Interaction

The M3I (Möller, Diewald, Roalter, & Kranz, 2014) is a framework that enables rich, context-driven multimodal interaction, oriented to the development of multimodal application for mobile devices such as smartphones. It allows the integration of modalities and rules that defines logical expressions and the action to be generated depending on the result of the expression. The

action of a rule can recursively call another rule or calls trigger activating the output modalities. Figure 2-19 presents an overview of the general structure of the framework. The structure contains a rule evaluator, the core of the framework, it processes the behavior of the system based on the defines rules.

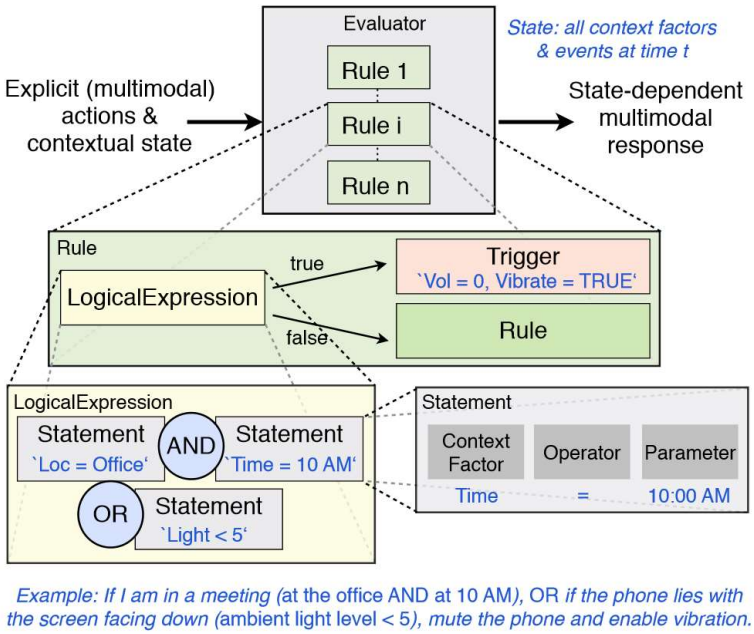

Figure 2-19 – Overview of the structure of the M3I framework.

2.5.10 Comparison of Multimodal Frameworks

Dumas et al. (2008) presented a comparison between some multimodal framework, focusing architecture, programming mechanisms and some characteristics (reproduced in Table 2-3). Watching the table is possible to observe many frameworks with different architectures, this way modalities created for a specific framework will not work in another framework. Only a few of the presented frameworks allow for extensibility, and fewer support the reuse of components. Even the HephaisTK, which supports extensibility and reuse of the components, since it uses the Java environment platform and agents for communication, it can face limitations when creating modalities for platforms that do not support Java. The referred table does not cover all the frameworks

presented in Section 2.5. For those, a discussion on their problems and limitations is presented in Section 2.7.

Table 2-3 - Comparison of different multimodal toolkits and architectures, from (Dumas, Lalanne, Guinard, et al., 2008)

	ICARE – OI [7]	OpenInterface [2]	IMBuilder/ MEngine [5]	Flippo et al. [8]	Krahnstoever [14]	Quickset [8]	Phidgets [8]	Papier-Mâché [8]	Java Swing MM Extension	Service Counter System	HephaisTK
Architecture traits											
Finite state machine			x							x	x
Components	x	x					x		x		
Software agents				x		x					x
Fusion by frames					x						x
Symbolic-statistical fusion						x					
Programming mechanisms											
Programming via "hard coding"					x	x				x	
Programming via API				x			x	x	x		
Programming via configuration file											x
Visual Programming tool	x	x	x								
Characteristics											
Extensibility		x	x	x					x	x	x
Pluggability							x		x		x
Reusable components	x	x									x
Open Source	x	x						x		x	x

Although many multimodal frameworks are available, practical software is not seen in the market for use by third parties. And this kind of frameworks will only become a valuable asset if they are used in new applications, to encourage developers to adopt and create new modalities.

2.6 Multi-device in Multimodal Interaction

Figure 2-20 shows a representation of the diversity of devices available, which can be explored its combined use in the context of multi-device scenarios.

A review of existing research literature showed that several works have been done in order to achieve interaction in multi-device scenarios. Common topics related to the multi-device are ubiquitous, migratory multimodal interfaces and very commonly they are more related to multi-display, not exploring the full potential of multimodal interaction.

Figure 2-20 – Representation of the diversity of devices and screen sizes. From left to right: desktop, laptop, tablet, smartphone.

In the area of ubiquitous multi-device, the work presented by Kernchen et al. (2010) focus on the steps needed to enable the adaptation of multimedia content and define a framework functionality to create applications. HIPerFace (Weibel et al., 2011) supports the integration of multiple devices, the defined architecture allows a rapid prototyping and experimentation with different devices. The architecture is based on three layers: the producers, interpreters and consumers. The Interactive Workspaces project (Johanson, Fox, & Winograd, 2002) explore a collaborative scenario for people to work together, allowing the interaction through multimodal devices, including mobile.

The work presented by S. Berti and F. Paternò (2005), describe the adoption of migratory multimodal interfaces, in which allow the users to switch from one device to other continuing the task that was performing before. In (Blumendorf, Roscher, & Albayrak, 2010), the work presented describes a multimodal system capable of adapting the user interface or the used modalities of the different devices, such as TV or smartphones, depending on the context.

The PolyChome (Badam & Elmqvist, 2014) is a web based application framework, which enables collaboration across multiple devices by sharing interaction events and managing the different displays. Since it is a web based framework, it shares across platform browser level events. Another similar solution is the Tandem Browsing Toolkit (Heikkinen, Goncalves, Kostakos, Elhart, & Ojala, 2014), which allows developers to rapidly create multi-display

enabled applications. Conductor (Hamilton & Wigdor, 2014) and VisPorter (Chung, North, Self, Chu, & Quek, 2014) are other examples of multi-display frameworks. The Thaddeus (Woźniak, Lischke, Schmidt, Zhao, & Fjeld, 2014) is a system, which enables information visualization for mobile devices.

The WATCHCONNECT (Houben & Marquardt, 2015) toolkit allows prototyping applications, which enables interaction through smartwatches with computers, mobiles or tables. This work presents a different way of interaction that uses the hardware capabilities of smartwatches.

In other works, some recommendations are made, to design and develop multimodal and multi-device application, such as in (Seyed, 2013), which presents a study to identify better interaction design for multiple displays, resulting in a set of guidelines to improve user experience. F. Paterno (2004) addresses and discusses relevant characteristics to considered in the process of designing this type of interfaces.

In (Shen, Esenther, Forlines, & Ryall, 2006) the author proposes three modes for multi-surface visualization and interaction, which are independent, reflective, and coordinated. In the first, devices work independently, while in the second each device shows the same content, and in the last it basically shows the same content but shows different viewpoints.

2.7 Discussion

A wide range of devices supporting multimodality are available nowadays at low cost. The diversity and complexity of the existing technologies supporting the modalities, as previously described, require the ability of the multimodal systems to deal with new technologies transparently, easily adapting to new modalities, and that developers are able to integrate new modalities without the knowledge of their internal complexity.

As shown, a number of tools and frameworks supporting multimodal interaction were proposed, each with different approaches, architectures, and ways to communicate among components. Since architectures and

communication are different, it makes it harder to adapt the interaction modalities to the different architectures. For some modalities, e.g., speech, if a developer wants to enable it for different systems he needs to deal with the complexity of each system, and probably implement a custom modality for each system. Given these limitations, only a limited number of modalities will be available for each framework and application. The lack of initial support for a basic set of modalities will prevent developers from fully adopting a framework as it poses a high entry cost and is far from enabling the full potential of multimodal applications. From the developer point of view, the adopted framework should already support a basic set of modalities, usable by any application adopting the framework, and allow easy addition of new modalities at any time.

The lack of a proper approach to the architectural aspects of multimodal systems is made clear by (Lawson et al., 2009) stating that there are solutions available that try to fill the gap between design, specification, and implementation of multimodal systems, but have problems:

1. Small or non-extensible number of devices
2. Platform/technology dependent
3. Not flexible to integrate with new devices or algorithms

Here is where solutions aligned with the W3C standards for MMI can be a valuable asset. These propose a new standard for multimodal architectures defining standard markup for communication and specifying a set of internal components of the architecture. Despite having some open points, such as discovery of new modalities in the system or its focus on web scenarios, it seems as a good starting point to create multimodal applications as it serves the principles of a decoupled architecture, enabling easy integration of modalities, and is open enough to encompass new evolutions. The standard was created aiming at web scenarios, namely web applications and web pages, but we argue that its use can be extended to other usage scenarios, which may be a perfect

match to the different challenges presented by multimodality in scenarios such as AAL. Such proposal is the subject of the next chapter.

2.8 Summary

This chapter presented an overview of the state-of-the-art for the different areas deem relevant for the objectives of this work. As made clear by the reviewed literature, existing work aiming to support the design and development of multimodal interaction systems is far from being able to address a number of important challenges. This precludes, for example, the consideration and further development of multimodal interaction solutions that are widely adopted and makes it difficult to harness the full potential of existing technologies. In light of this evidence, the next chapter proposes a novel framework to support multimodal interaction addressing these issues.

Chapter 3

Framework for Multimodal Interaction

In line with the current state-of-the-art and the challenges identified in the previous chapter, this proposal for a multimodal architecture aims to fill the gaps found in other multimodal architectures, allowing an easier integration in new applications.

The following proposed architecture was developed over several iterations. In each iteration the architecture has evolved by fulfilling the requirements determined at each stage, and meeting, at the end, all the objectives of this work. Overall, we aimed to create a framework to simplify the integration of multimodal capabilities into new applications, allowing developers to focus on other content of applications. In line with the advantages identified in the previous chapter, an effort has been made to propose a framework aligned with the concepts and recommendations of the W3C for a multimodal architecture. The next sections present the usage areas, requirements, architecture and modules.

3.1 Usage Area

The proposal for the multimodal interaction architecture and framework aims to cover a broad set of scenarios, but Ambient Assisted Living (AAL) was given a special importance. In AAL scenarios, traditional interaction methods such as mouse and keyboard are not fit to some users and use contexts but, for instance, speech or eye gaze might provide the means to enable such users to interact with applications.

Developing applications for this kind of scenarios, either for elderly people or for people with special needs, presents a great effort and many aspects need to be taken into consideration. The diversity of users and their needs requires the consideration of a wide set of interaction methods and their subsequent implementation for each application.

In our proposal, one application should provide features to enable their usage in different contexts and to adapt to the diversity of users. The possibility of using multiple interaction modalities in one application, exchange modalities by others that suit better a particular user, and the use of interaction designs that feel more natural to users, translates to a more engaging and pervasive application. This also opens the possibility to create applications to serve a broader set of users, even users that are not familiarized with technologies.

3.2 Requirements

Based on the existing solutions, described in the previous chapter, and in the objectives of this work, a set of requirements were identified that should enable a simplified method to develop multimodal applications. To start, the solution of a multimodal framework and its consideration to support a new application should be **clean and easy to adopt**. The framework **must implement a loosely coupled architecture**, providing developers with the flexibility to add, remove or change modules while ensuring that the system, as a whole, continues to work seamlessly without even knowing that a module was changed. This

loosely coupled architecture potentially enables an easier and better integration with new applications.

The framework should be based **on languages and protocols adopted as international standards**, since many developers already have knowledge of the standards, enabling an easier understanding of the framework and shorter learning periods, either to develop applications or to create new modalities. This brings an important feature that should be included in the multimodal framework, **the possibility to be extended**.

Since multi-device interaction is one of the objectives of the work, the framework must enable this scenario**, supporting the distribution of modalities across multiple devices** such as PCs, tablets, smartphones, etc., taking advantage of the technologies from each device to enable new modalities. Also, it should support the **management of multiple interaction managers in different devices**.

The multimodal framework should **provide multiple input and output modalities that should be as generic as possible** to enable that they are used and easily adopted in new applications supporting multimodal interaction. As speech is one important way of communication between humans, and it can be one of the most difficult modalities to develop and integrate due to language dependence, some special attention should be given to the **speech modality**. This modality must offer features that **enable dealing with multiple languages** and seamless deployment of applications producing uniform outputs regardless of the targeted language.

In the desired application scenarios, such as the overall scenario presented in the previous sub-section, **modalities should always be active, ready for the interaction with the user**. For instance, speech recognition must be ready for any command of the user and ready to transmit the event.

3.3 Architecture overview

The proposed architecture follows some aspects of the standard defined by the W3C, which already fulfills some of the initial requirements. Nevertheless, while the Multimodal Interaction Architecture (MMI Arch), proposed by the W3C, only addresses web contexts, our proposal aims to cover a broader set of scenarios such as desktop or mobile applications.

Inheriting from the decoupled nature of the W3C architecture, our proposal is also divided in four major components and adopts the markup languages used to transport notifications between components. The architecture acts similarly to a Model Viewer Controller (MVC) paradigm where the components, as illustrated in Figure 3-1, are:

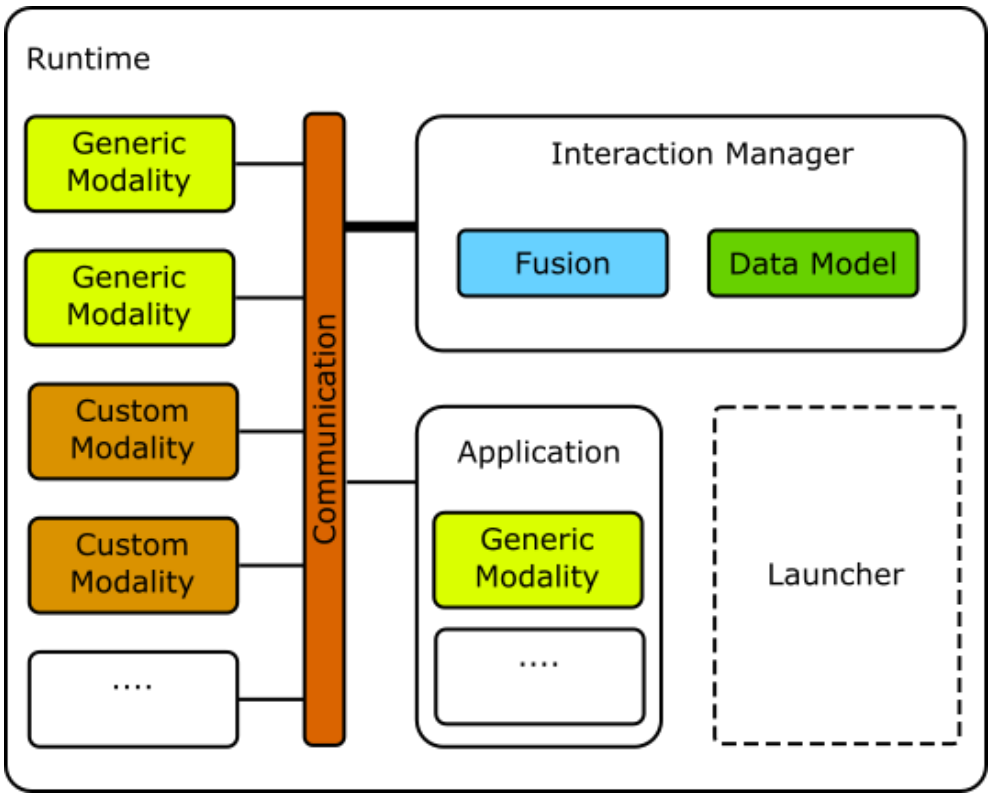

Figure 3-1 – Proposed Multimodal Framework Architecture

- *Runtime Framework* – defines all the services needed to execute multimodal applications. An important feature of the Runtime is the

definition of the event transport layer, the communication protocol and markup languages for the communication between interaction manager and modalities. Also, it provides the necessary tools to start all needed modules for multimodal interaction.

- *Interaction Manager* – it is one of the most important modules of the multimodal framework. The interaction manager is responsible for managing all user's interaction with the system. It receives all events generated by the input modalities and produces new messages to be delivered to the output modalities. All events and messages have to go through the interaction manager. Internally, the interaction manager processes the events using a state machine, which can be configured using SCXML.

- *Fusion* – fusion of events in multimodal interaction is an important topic, in which multiple interaction events can be combined into new single events that aggregate all the information provided in the events received from modalities. Fusion, in our architecture, is a part of the interaction manager, but decoupled from its core, to allow the exchange of the module by other fusion modules.

- *Data Model* – The interaction manager needs to store information regarding state, context or last events. This is performed using a data component, also responsible for managing the persistence of data during the runtime of the interaction manager.

- *Modality Components* – Each modality in the system is considered a component of the framework, and it is either an input or output modality. Modalities with input capabilities are responsible for listening to users' commands and generating events; modalities with

output capabilities provide ways for the system to deliver contents to the user. Noting that a modality, in the definition of the W3C, can be more complex, and does not have to be limited to input or output, one can broadly describe it as the interface between a human and the multimodal interaction system.

We defined generic modalities and custom modalities, the first are mainly the modalities provided in the proposed framework, aiming to serve a wide set of applications. Custom modalities are more specific modalities, generally adapted to a particular context.

Communication between modules, the interaction manager and modalities, implements the W3C multimodal architecture life cycle events. These events provide a standard communication language for the interaction manager to invoke the modalities and receive events.

3.3.1 Enhanced architecture to Support Multi-device

In order to support multi-device applications with multimodal interaction capabilities two approaches were taken. In the first approach, each device runs an instance of the interaction manager while, in the second approach, a cloud based interaction manager runs remotely and each device connects to the same interaction manager.

In the scenario where each device runs an interaction manager, illustrated in Figure 3-2, the modalities of each device only communicate with the interaction manager of that device. Both devices can run the application, meaning that the modality of the interface must run in both devices and other modalities can run in each device according to the users' preferences and device capabilities. In this scenario, mechanisms are implemented to allow the identification of the interaction manager in a local network, and each interaction manager behaves as a modality to the other interaction manager. Following this

approach, it is possible to disconnect the two devices and work with each device separately.

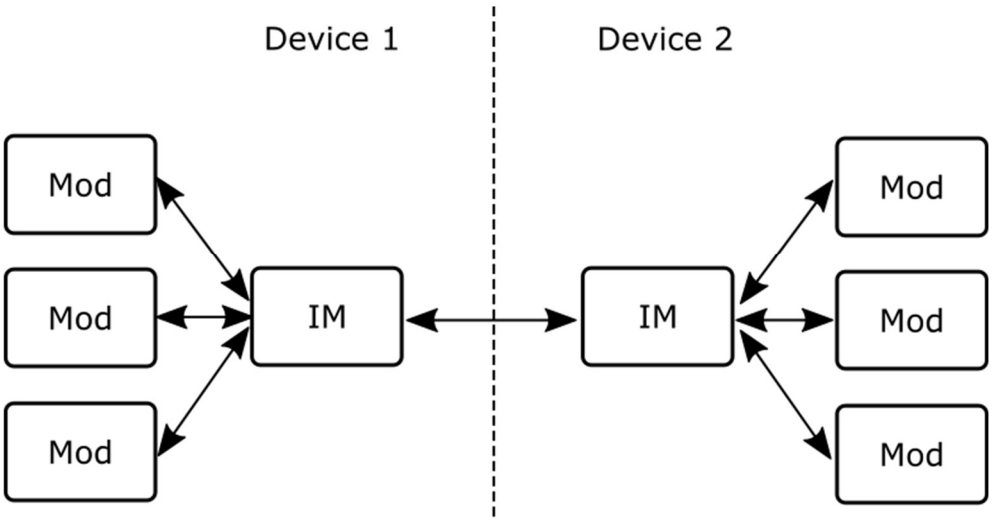

Figure 3-2 – Multi-device architecture, each device runs an instance of the interaction manager

In the second approach, only one instance of the interaction manager is running. All modalities of the multimodal systems know the location of the interaction manager and communicate with it. In this scenario, depicted in the diagram presented in Figure 3-3, it is simpler to add more devices to the system.

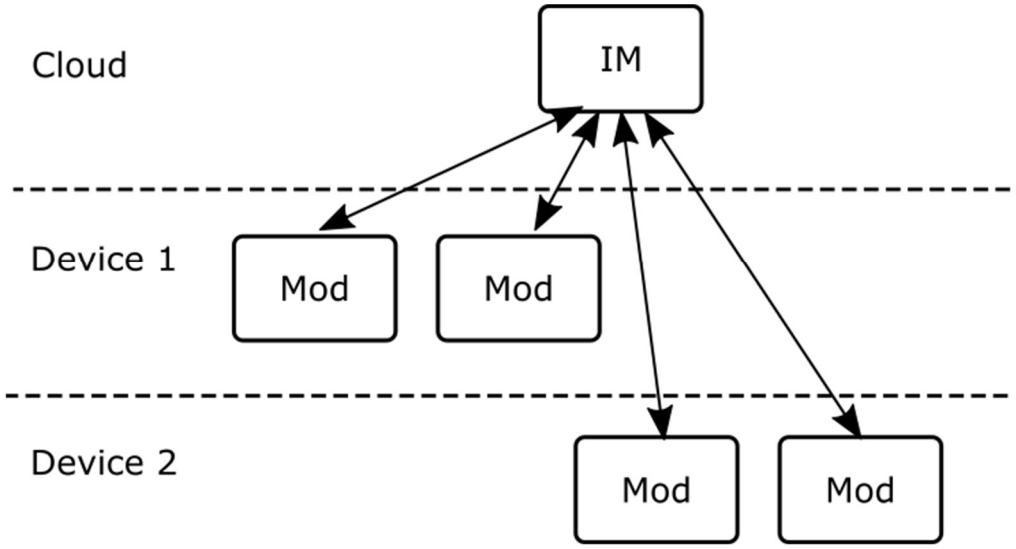

Figure 3-3 – Multi-device architecture based on a cloud interaction manager

3.4 Multimodal Framework

All the components of the multimodal framework run independently from each other and communicate using well-defined languages and protocols. This section discusses the purpose and responsibilities of each module, their basic functioning and how they communicate through the interaction manager.

3.4.1 Runtime Framework

Although this module of the framework has a more conceptual part, defining how the communication is performed, in the system, it is also responsible for starting each module of the multimodal framework.

The communication, as further explained in the section detailing the interaction manager, is performed using the HTTP GET or POST protocols and the messages' content is encapsulated in the markup language of the life cycle events.

In our approach, the most important life cycle events are the *NewContextRequest* and *NewContextResponse* that, in addition to requesting the context of the application, also notify the interaction manager of the availability of a particular modality. The *extensionNotification* event encapsulates the recognized event from the input modality and informs the interaction manager of its occurrence. Each generated event is described using the markup language EMMA, containing all relevant information about the event, including timestamps of the beginning and end of that event. This life cycle event can be generated by the modalities or interaction manager. Finally, *StartRequest* and *StartResponse* are life cycle events used by the interaction manager to communicate and invoke actions in the modality components. The *StartRequest* contains a *Data* element to transport the information to the modalities.

In Figure 3-4, an example is presented showing a sequence diagram describing the life-cycle events exchanged between the interaction manager and

a set of modalities available in one multimodal system. The diagram shows the events when the modalities start, when the speech modality generates a speech event and sends the notification to the interaction manager, and the invocation of the GUI modality by the interaction manager.

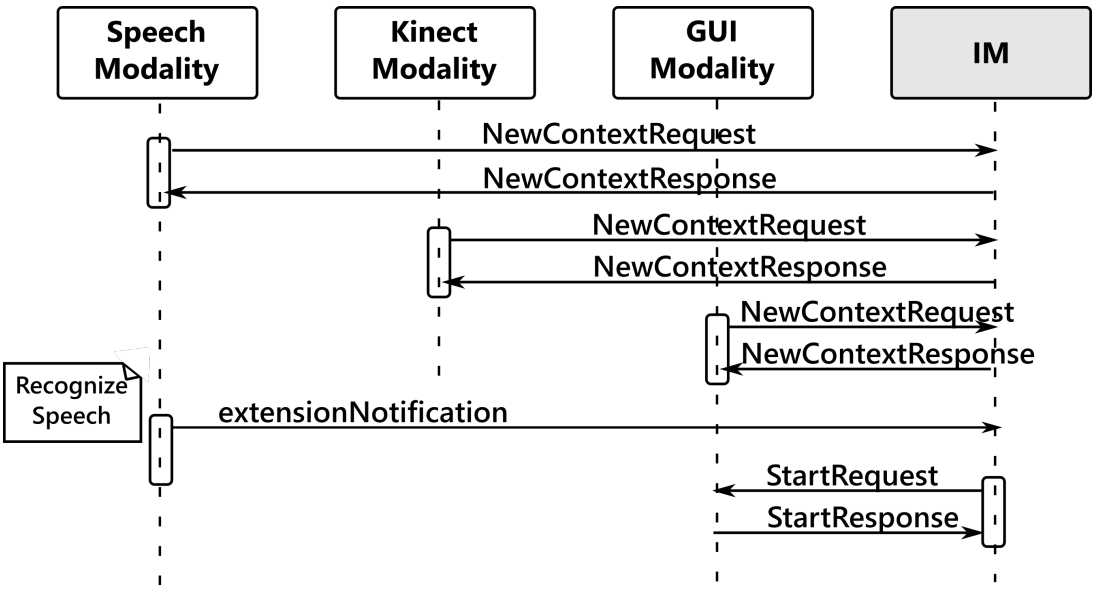

Figure 3-4 - Sequence diagram presenting life cycle events between interaction manager and modalities

In the example diagram, illustrated in Figure 3-5, it is possible to identify the communication between modalities and interaction manager and between the two interaction managers. When the first event is generated by the touch modality, in Device 2, it is sent to the interaction manager and it resends that event to the other interaction manager in Device 1. After that, depending on the state, data model and modalities available of each device, they process the event and proceed accordingly.

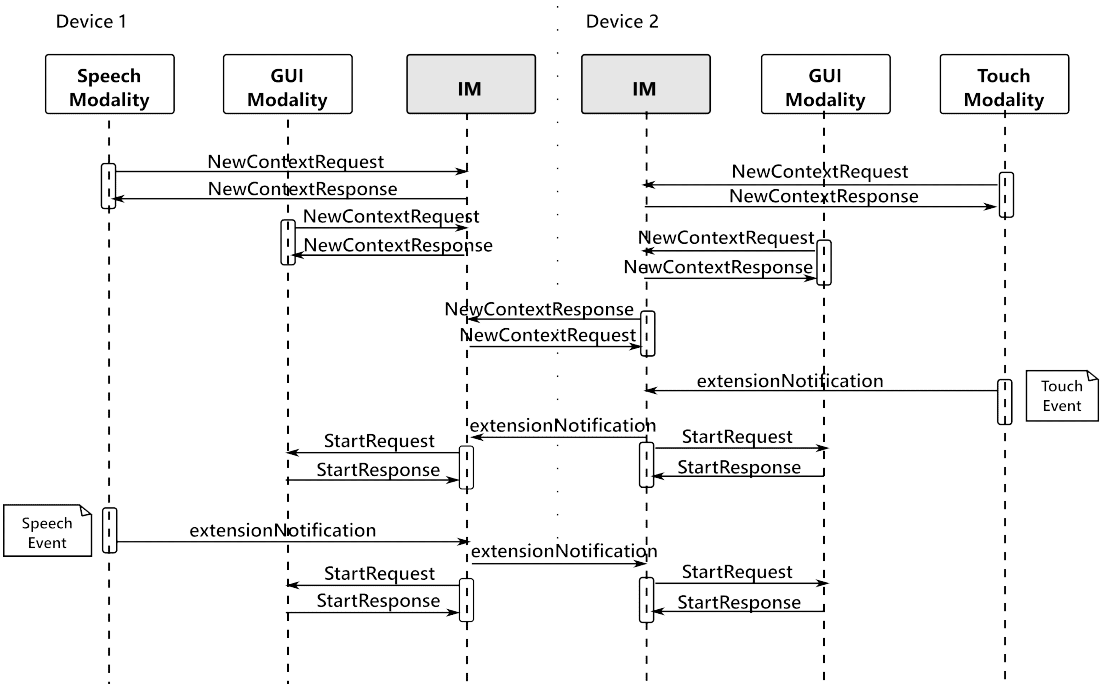

Figure 3-5 - Sequence diagram of multi-device interaction, presenting life cycle events between interaction manager and modalities, and the two interaction managers.

The other part of the Runtime module is a launcher responsible for starting the default modules and modalities. It starts by running the interaction manager, then all the modalities. It also monitors the activity of the modules and, in case of any failure, the modules are restarted.

3.4.2 Interaction Manager

The interaction manager is one of the most important modules of the framework and it manages all life cycle events coming from a number of modalities. The interaction manager has an integrated data model that enables storing some environment context of the user and the system. In our scenario it is expected that each running modality is always waiting for events or life cycle events from the interaction manager.

When a modality starts to run, it signals the interaction manager using the *NewContextRequest* life cycle event and the interaction manager confirms the reception of the request by responding with a *NewContextResponse*. After the

interaction manager receives the life cycle events it registers that a new modality is available, in the data model, so it knows that new life cycle events can be sent to it, in the future. At all time, the interaction manager knows the available modalities with which it can communicate.

The interaction manager is capable of sending the *StartResquest* life cycle event, to any modality available and the main objective of this event is to deliver information so it can be presented to the user. Modalities must be aware of the markup used in these events to interpret and present the information to the user. When one modality receives a *StartRequest*, it must respond with a *StartResponse* to notify that it has received that life cycle event.

The next most important life cycle event is the *ExtensionNotification*, which is used by the modalities to deliver the event information to the interaction manager. These messages are processed by the interaction manager that, depending on the source of the message and the type of the message, can generate one or more new *StartRequests* and send it to the corresponding modalities. The *ExtensionNotification* exempts the interaction manager to send a response.

Since, in our scenario, all modalities are always ready, the number of used life cycle events was reduced, for simplification purposes. For instance, the *PrepareRequest* life cycle event was not considered while developing the interaction manager, yet the life cycle is implemented and can be used in the future.

In our approach to the multimodal framework, modalities know where the interaction manager is located. The communication is initiated by modalities using the HTTP protocol, even when it is the interaction manager that wants to send requests to modalities. It considers both HTTP GET and POST requests for the communication. This way, when modalities wants to send a request or notification to the interaction manager it is done using HTTP POST, to receive requests from the interaction manager it is done using HTTP GET. In the last

case it forces each modality to poll the interaction manager for new requests but, in this way, it does not require for each modality to implement an HTTP server.

In order to extend the use of the multimodal framework to work in a multi-device scenario, the interaction manager was provided with several of the features of a modality. One interaction manager will be seen as a modality to the other interaction manager and each one will run in a different device. It is capable of transmitting *ExtensionNotification* events, but only to other interaction managers. The communication between the two is also done with the HTTP protocol, with the difference that, in the beginning, they do not know the location of the other. For the two interaction managers to acknowledge the existence of the other the protocol Simple Service Discovery Protocol (SSDP) was used. Each interaction manager runs the service and, when they start, they try to locate the other service: when one interaction manager finds another they exchange their address, so they can communicate via HTTP protocol.

3.4.3 Fusion

The main objective of our fusion module is to simplify the process of creating and including fusion of events in new applications, removing some of the complexity when designing multimodal interaction.

As previously explained, in our approach the fusion module is seen as part of the interaction manager, but still preserving a decoupled nature, as presented in Figure 3-6. This way, the module can be replaced by other fusion module or even removed from the system. Basically, the module works similarly to the interaction manager and the core is still a SCXML state machine, which receives events and sends events to the main interaction manager. A particular aspect that should be highlighted as an important focus of attention is the method to configure the fusion engine supporting the creation of the corresponding SCXML. In a system with many events and fusion points, the creation of that SCXML is complex and we considered that some method should be provided to enable its definition at a higher (more conceptual) level.

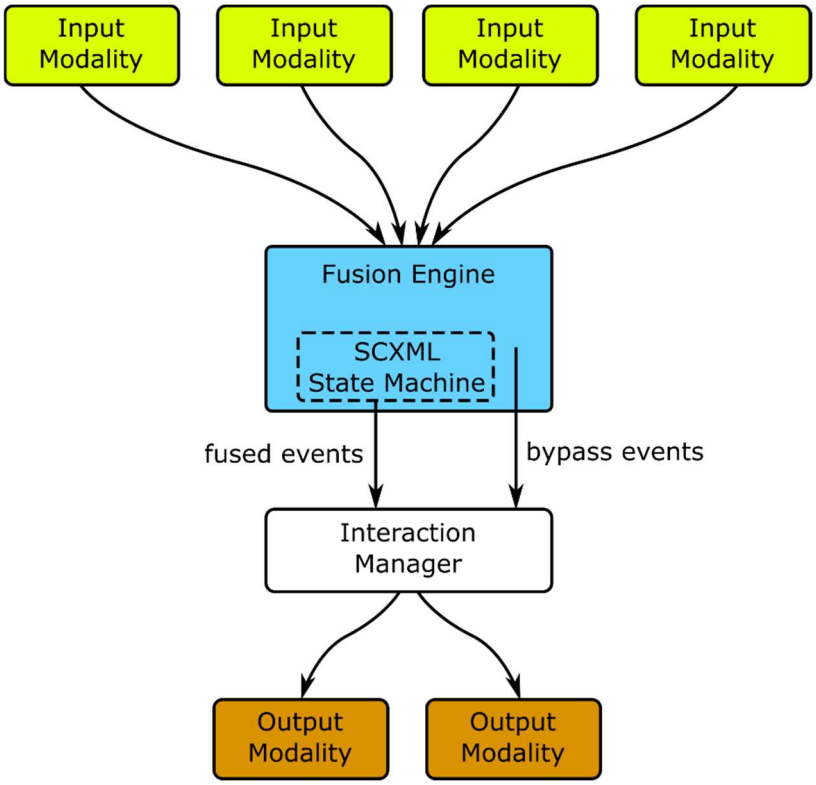

Figure 3-6 - Multimodal Fusion Architecture. Input modalities send events to the fusion engine, the events can be fused into new ones or the fusion engine can directly pass them to the interaction manager. Then, the interaction manager interprets the events from modalities and the fused events.

In our method, all modalities must publish all events that they can generate by creating a file with a predefined structure. Also, a file with the same structure must be created to define the output events. The generation of those files is done according to the syntax of a programming language, enumerating all the relevant information. Additionally, a class is proposed supporting the different operations required for fusion, enabling the definition, by the developer, of which events to fuse and how. By importing the files generated for each modality to an integrated development environment (IDE), all the features of the IDE are available, such as autocomplete or syntax suggestions. These features help the developer to create a set of lines of code that are a high-level description of the intended combination of events. Compiling that code automatically generates the SCXML file to configure the fusion engine.

In order to have coherent events, they must be well defined. Defining a semantic layer based on dialog acts (Bunt et al., 2010) and extending the use for every modality, even for non-speech related modalities, results in the standardization of the semantic output. For instance, a swipe left touch event and speech event of "turn left" can produce the same output semantics. These options will reduce the complexity of defining the fusion.

A dialog act is mainly defined by two components (Bunt et al., 2010), the communicative function and the semantic content.

3.4.4 Modalities

Our overall approach for modalities is aligned with the concepts proposed by the standard W3C multimodal architecture. Modalities can be complex and include features of different subjects. The proposed framework includes some generic modalities, which can be used in different scenarios and applications. Offering a diversity of modalities allows developers to have only the concern of developing the main application. Distributing a multimodal framework without generic modalities would result in the non-adoption of the framework, due to the extra effort required to develop new modalities and the technical knowledge needed to create them. Although modalities are decoupled from each other, the Graphical User Interface (GUI) Modality and Touch modality are somehow included in the application, the GUI Modality because it is responsible for changing the information displayed by the application and the touch modality to enable it to be aware of the objects displayed by the application. Speech, gestures and eye-gaze run completely independent from the application.

3.4.4.1 Generic Speech Modality

Due to its relevance for our main application area (AAL) a generic speech modality is part of our framework. This modality features speech input, i.e., automatic speech recognition (ASR) and speech output, i.e., speech synthesis. The language dependency of speech related modalities is one of the main

challenges to develop a generic speech modality. In the case of speech recognition, models and grammars must exist for each desired language. Our overall purpose was to provide a generic speech modality that supported, as best as possible, the development of Multilanguage applications. Figure 3-7 presents all the modules included in the generic speech modality. It is composed by three main modules: (1) a general translation service, which translates written text; (2) a grammar translation service, responsible for translating a grammar and keep it in the same format, manage those grammars and extract the meaning of recognized sentences; and (3) a part that runs locally, which includes the speech engines for recognition (input) and synthesis (output).

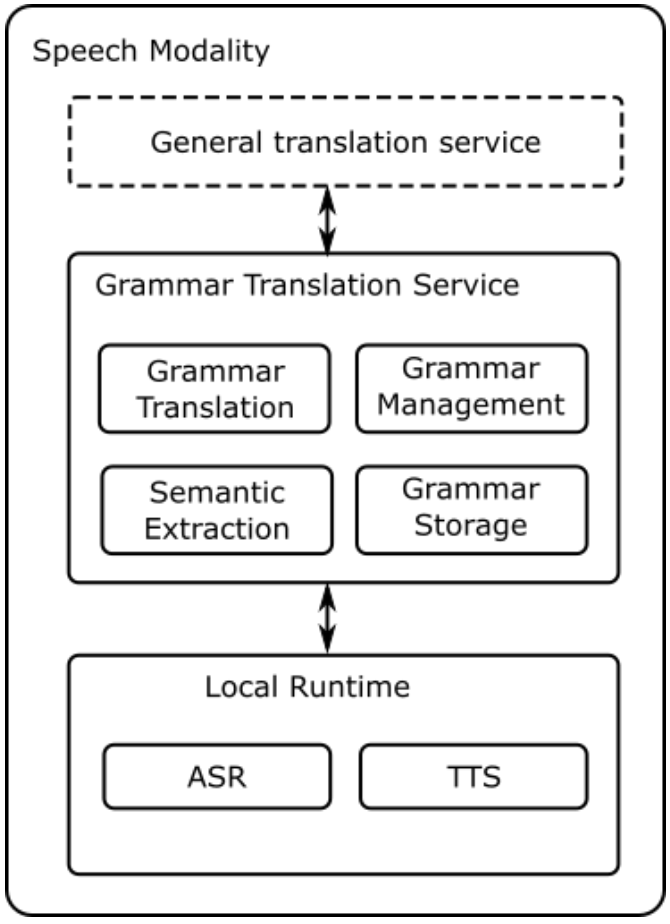

Figure 3-7 – Overview of the generic speech modality modules, from bottom to top: the local runtime with ASR and TTS features, grammar translation service and a generic translation service.

Speech Input – The speech modality supports two different ways of recognition, the recognition of commands and dictation. The first is supported by a speech grammar and the second by a dictation model using a statistical language model. The modality is capable of switching from recognition of commands to dictation in runtime, whenever it receives a request from the interaction manager.

When the modality is loaded and sends the *NewContextRequest* life cycle event to the interaction manager it receives the context of the application and, with that, the current language of the application. Then, it must be configured with the corresponding grammar for that language. The proposed solution to tackle the multilingual challenge was the creation of a cloud-based translation service (Teixeira et al., 2014; Teixeira, Francisco, Almeida, Pereira, & Silva, 2015). Figure 3-8 shows the architecture and communication between the modules. The service is capable of automatically translating an English grammar, uploaded by the developer while developing the application, into the target languages. Those newly, automatically created grammars for the other languages can be reviewed, through a dedicated webpage, corrected and expanded with more sentences with the same meaning. The service also supports the inclusion of dynamic rules so that the grammars can be changed in runtime. This enables the recognition of specific dynamic content of the application.

Whenever the user interacts with the modality and speaks any sentence that is recognized by the speech engine, the modality requests the translation service to extract the meaning of the sentence, its semantic information. As previously discussed, the semantic information is defined with a semantic layer based on dialog acts and the produced semantic output will be the same regardless of the input language.

Table 3-1 presents an example of the extracted semantics using the sentence "Show schedule Monday" and the Portuguese translation "Ver calendário Segunda-feira", both produce the same output act, which in the application will result in the same action.

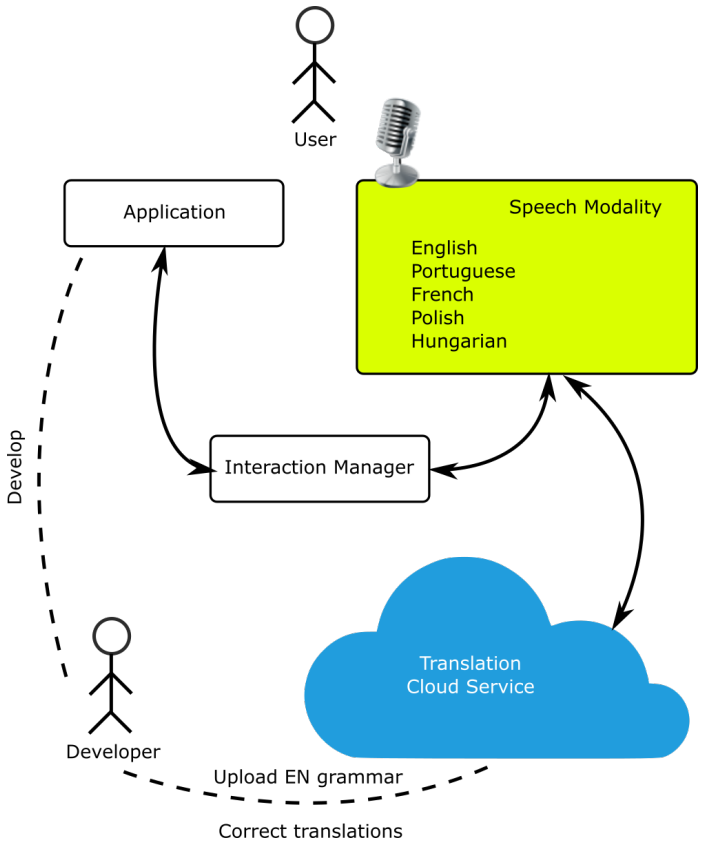

Figure 3-8 - Architecture of the speech input modality in the context of a multimodal system

Table 3-1 - Example of the result of the semantic extraction performed by the cloud-based service

ACT	**[Main] [OPEN]**	**[Main] [OPEN] [AGENDA]**	**[Main] [OPEN] [AGENDA] [WEEKDAY] [MONDAY]**
Portuguese	Ver	calendário	Segunda-feira
English	Show	schedule	Monday

Speech Output – The supported languages of the speech output are the same of the supported by the speech input. When the multimodal system needs to transmit a message to the user through speech, it sends a message to the modality encoded in Speech Synthesis Markup Language (SSML). This markup language supports the encoding of the content and other speech related parameters. To avoid the recognition of synthesized speech, the speech input is put on hold while the message is played to the user. To synthesize the speech a speech engine was used that already has voices for several languages, but most of the languages were limited to one voice. For European Portuguese additional voices were recorded and trained (Almeida et al., 2015), giving users the possibility to select their preferred voice.

3.4.4.2 Other generic modalities

Besides the generic speech modality the framework must also include generic modalities for other commonly used ways of interaction. The base set of modalities includes gestures, eye-gaze, touch and classical graphical user interface (GUI). These modalities are briefly presented in what follows.

3.4.4.2.1 Gestures Modality

The gestures modality is a simple modality, which recognizes some of the users' body gestures. Using Kinect technology, it is possible to track a person's skeleton, recognize the hand and follow the movement of the bone joints. The recognized movements are the hand swipes (right, left, up and down), and its use is particularly interesting if the user wants to scroll some content displayed in the screen and is away from other input device or is not in range for speech commands.

3.4.4.2.2 Eye gaze Modality

The eye gaze modality detects the gaze direction and identifies the location, on screen, to where the user is looking at. To retrieve this information a specific device built for this purpose is needed (e.g., eye tracker). This modality is mainly

used along with the speech modality fusing the events of both modalities. For instance, if a user wants to select an object he can look at it and say “select”.

3.4.4.2.3 Touch Modality

As happens with the GUI modality, the touch modality also works tightly coupled to the application. The actions of buttons or other elements must call a method that sends the input event to the interaction manager.

3.4.4.2.4 GUI Modality

The GUI modality is responsible for dealing with the displayed content, behaving as an output modality. In some cases it can behave as an input modality, such as when the application has some information that can be presented in a different medium and it sends a notification to the interaction manager, for instance, written text that can be synthesized. Since this module runs inside the application, the developer of the application must deal with it. We provide a library to include in the application which implements the life cycle events and communication with the interaction manager. The developer only has to use it to receive and send life cycle events.

3.4.4.3 Custom Modalities

Other modalities can be added by developers. If communication protocols, markup language, and the messages semantics are followed any system will continue working seamlessly with the new modality. Since new devices and technologies emerge every day, supporting an easy way to integrate them into the multimodal framework is, in our view, of major importance.

3.5 Summary

The different issues and challenges identified in the literature are, from our perspective, significant barriers to attain an adequate context to design and develop multimodal interaction applications. This chapter starts from the

existing panorama and establishes a set of requirements that should be considered to foster improved multimodal interaction. These guide the proposal of a novel architecture and framework to support the design and development of multimodal interaction systems. While this chapter discussed the more conceptual aspects of the proposed framework, the next chapter will provide additional detail on the different technologies supporting its instantiation.

Chapter 4

Development of the Framework

In line with the previous chapter, the multimodal framework was developed piece by piece. The description of the implementation is divided into two parts: the basic infrastructure modules of the framework and the interaction modalities. The first includes the modules that are always needed, such as the interaction manager, and the second includes the generic modalities that are included with the multimodal framework. Each module is decoupled from the remaining modules and final users can choose their preferred modalities to use. The provided set of modalities covers some of the most used modalities but, at any time, given the decoupled nature of the framework architecture, other modalities can easily be added.

4.1 Basic Infrastructure Modules of the Framework

The development of the multimodal framework was structured according to the different modules of the framework. After having the first concepts and

architecture defined, the development work started with the implementation of the first version of the interaction manager and some basic modalities in order to test it. Since the method to define the architecture was iterative, the framework evolved until it reached its current version.

In the first prototype, the framework allowed the communication between the basic modalities and a prototype application. The communication was accomplished by using the HTTP protocol, the life cycle events and EMMA, the markup language used to transmit data in the W3C standards. In the first prototype, which served to test our concept, a basic speech modality, touch modality and an example of an application was developed. The interaction manager and these modalities helped to test and understand better the importance of multimodal interaction. Over a number of iterations, the interaction manager was improved and harnessed with more features, one of the latest being the capability to handle multi device scenario, by either having one interaction manager in each device or by considering a single interaction manager residing on the cloud. The generic modalities have also improved over time and the speech modality is the one that attained a higher complexity and maturity.

The development of the multimodal framework was linked to a number of R&D projects which, in addition to contributing to the requirements, resulted in a number of applications that use the multimodal framework and provide a richer scenario to show its capabilities and usefulness.

4.1.1 Interaction Manager (IM)

The interaction manager is a core module of the multimodal framework, responsible for connecting all other modules. The interaction manager is an application developed in Java that implements an HTTP server to receive events from input modalities. The events, in the form of markup language, are parsed and processed by the interaction manager using a state chart machine, defined using SCXML, adhering to the W3C recommendation. For this part of the

interaction manager we considered the Commons SCXML[5] library, from Apache Commons, and extended it to implement the desired instructions for some nodes of the SCXML such as the *send* node. The SCXML is also used to manage and store the Data Model component presented in the multimodal architecture. The Data Model can be used to store information related to the state of the application, but also to store a list of the available modalities so that, when the application needs to send any action, it knows the available modalities.

Figure 4-1 presents the internal components of the interaction manager. The main instance of the interaction manager initiates the SCXML state machine and an HTTP server that contains one handler for POST messages and another for GET messages. When an event is received by the server, the POST handler parses the event and triggers the state machine. The Function Mapper contains the methods that can be invoked from the SCXML state machine. Sending events generated in the state machine to modalities is the responsibility of the MMI Events Dispatcher, which sends a request with the life cycle event if the modality has a HTTP server, or simply responds to the polling of the modalities. When modalities poll the interaction manager, the GET handler waits for a timeout to respond with a renew message. If an event needs to be sent to that modality the GET handler passes the response stream to the MMI Events Distacher.

The basic SCXML used to configure the interaction manager includes two parallel main states, one to receive *newContextRequest* from modalities and set the modality to available state, in that context, and other to receive the notification of events resulting from user interaction.

Sending messages to modalities can be done in two different ways: if the selected modality has an HTTP server, the interaction manager sends the message directly; if the modality does not implement an HTTP server, the

[5] http://commons.apache.org/proper/commons-scxml/

interaction manager expects to be periodically polled by the modality for messages.

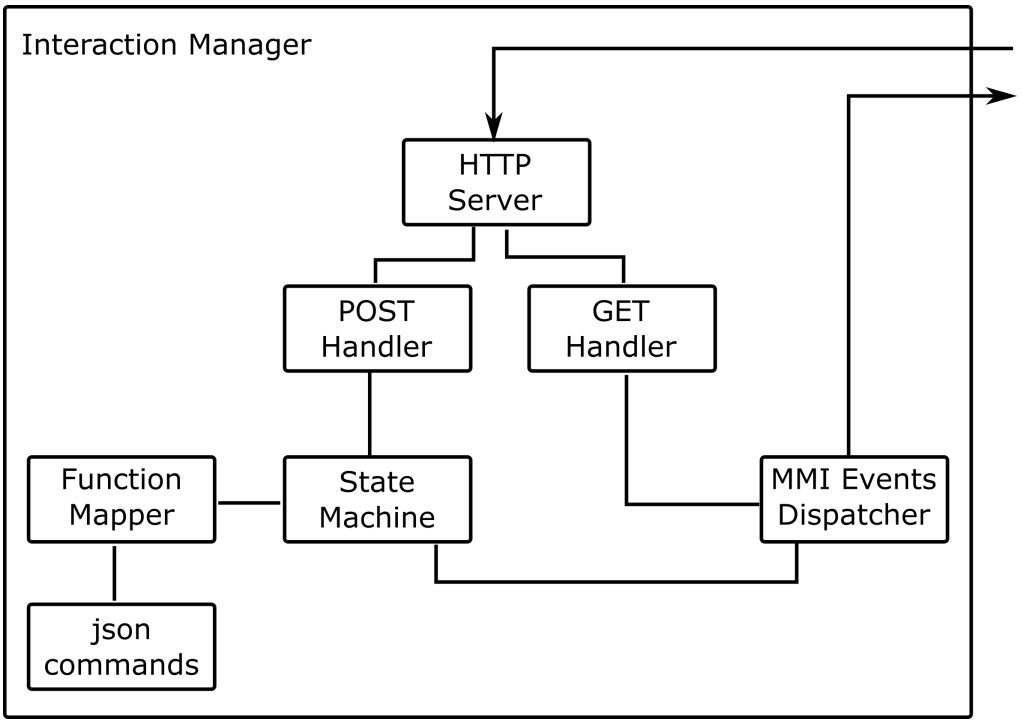

Figure 4-1 – Main internal components of the Interaction Manager.

In the previous chapter, in Figure 3-4, a sequence diagram was presented showing the life cycle events exchanged at start and resulting from an event occurrence, in the speech modality. Figure 4-2 presents the same situation but with additional messages regarding the synchronization of the HTTP requests and responses. It is observable, in the sequence diagram, the polling done to the interaction manager by the GUI modality. Each HTTP GET stays active for a defined period until it reaches a timeout and the interaction manager responds with a renew message so that the GUI modality can make a new request. When the interaction manager responds to the GUI modality with a *StartRequest* the GUI modality responds through an HTTP POST, to which the interaction manager simply responds with an empty response.

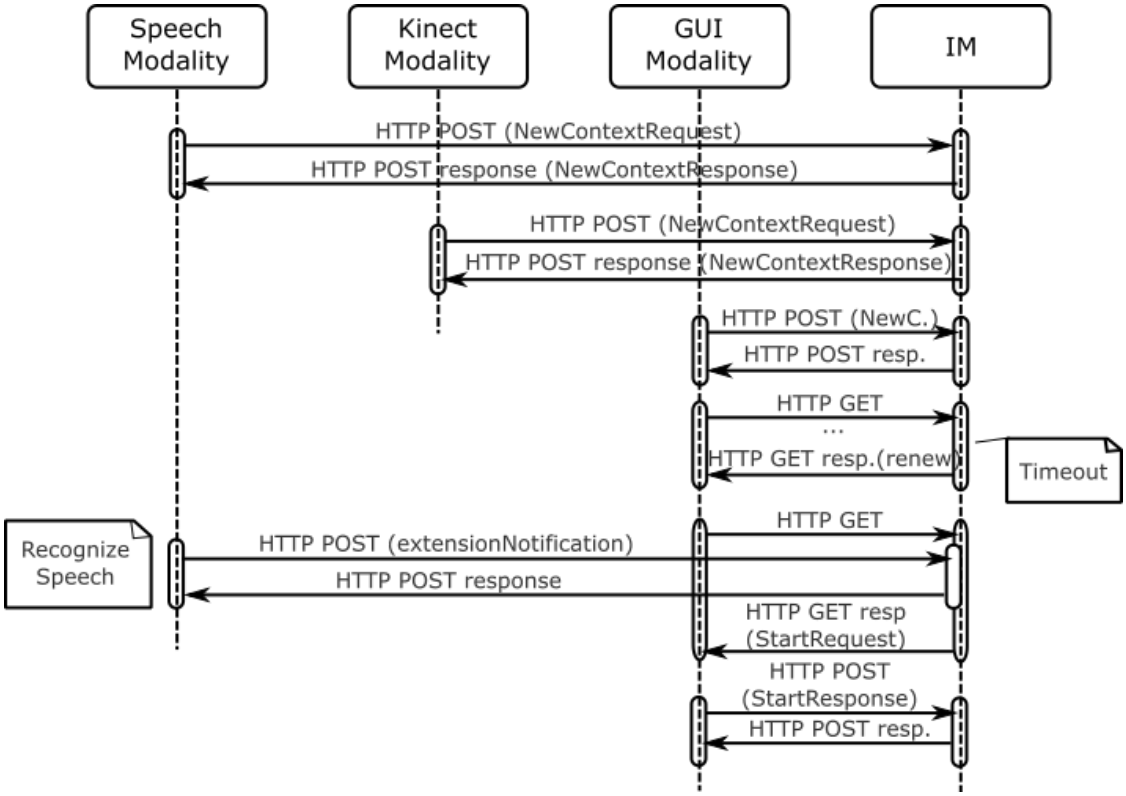

Figure 4-2 - Sequence diagram of the HTTP messages associated with the life cycle events exchanged between modalities and interaction manager.

4.1.1.1 Extending support for multi-device

In the extension for multi-device support, two approaches were taken: 1) each device runs an instance of the interaction manager and they connect to each other; 2) all devices connect to a single cloud-based interaction manger.

The first approach was implemented including, in the interaction manager, an UPnP Server in order to enable the discovery of the other devices. The service listens to broadcast requests, and has a service allowing for two interaction managers to exchange their local addresses. When the interaction managers start they send a broadcast message, trying to find other interaction managers. If it is the only one running it waits, with the UPnP server running, for other interaction managers. If there is one running, their addresses are exchanged and they can now start exchanging life cycle events. In this scenario, each

interaction manager only manages the input/output modalities that are available in that device, but sees the other interaction manager as an input/output modality. It works seamlessly with either one or two devices present in the system and might easily be expanded to encompass more devices.

In the cloud-based approach only one interaction manager is responsible for managing all the modalities available in the included devices. It has a very similar implementation to the one device scenario, but all modalities must poll the interaction manager in order to receive requests. This interaction manager, in addition to handling all the modalities, also supports multiple clients and each exchanged message must be associated with a client identification. This way, each running instance of the interaction manger is capable of running multiple state machines, one for each client.

Each approach to multi-device multimodal interaction has its advantages, and both are available in the multimodal framework to enable the choice, by the developer, of the best suited for the envisaged scenario. In the first, the interaction manager only manages the modalities of the local device, so it is easier to use proximity based modalities to control the information to present or to choose the output modality. If the devices are running in the same local network, they find each other easily and there is no need to identify the client. In the second approach, it is easy to manage more devices and devices do not need to be in the same network.

4.1.2 Fusion

The fusion module is, in our architecture, conceptually part of the interaction manager, but it is implemented as a separated module. This way, this module can be changed and the rest of the system remains the same. The core of the module is a state machine, defined using SCXML, in a very similar way to what is done for the interaction manager.

The concern when developing this module was to devise a way to simplify how the fusion state-machine could be configured, by providing a systematic method to define the corresponding SCXML configuration file. In this context, it was defined that each modality must publish all the possible events that can be generated in its context. The events, input and output, must be in the format of a Java *enum* as in the example shown in Code 4-1 (simple touch modality). Events are defined by their name and a timeout. The developer only has to create a file defining the rules to fuse events.

```
1  import scxmlgen.interfaces.Modality;
2
3  public enum Touch implements Modality{
4
5      OBJECT_A("OBJECT_A", 1500),
6      OBJECT_B("OBJECT_B", 1500);
7
8      private String event;
9      private int timeout;
10
11     Touch(String m, int time) {
12         event=m;
13         timeout=time;
14     }
15
16     @Override
17     public int getTimeOut() {
18         return timeout;
19     }
20
21     @Override
22     public String getEventName() {
23         return getModalityName()+"."+event;
24     }
25
26     @Override
27     public String getEvName() {
28         return getModalityName().toLowerCase()+event.toLowerCase();
29     }
30 }
```

Code 4-1- Example of a generated Java enum, by the touch modality. Lists the set of possible events (object_a and object_b) and implements basic functions to be used in Java.

Using the *enums* and a provided API, it is easy to the developer to generate an SCXML file. A class, called FusionGenerator, is used to create the SCXML

file, and it includes the methods to describe the type of fusion: complementary, redundancy or single. Figure 4-3 shows the representation of the state machines for the complementary and redundancy types. The single type only has a transition and a state that immediately forwards the event to the interaction manager. Complementary constrains the order of events, but the combination of two complementary rules can overcome this limitation.

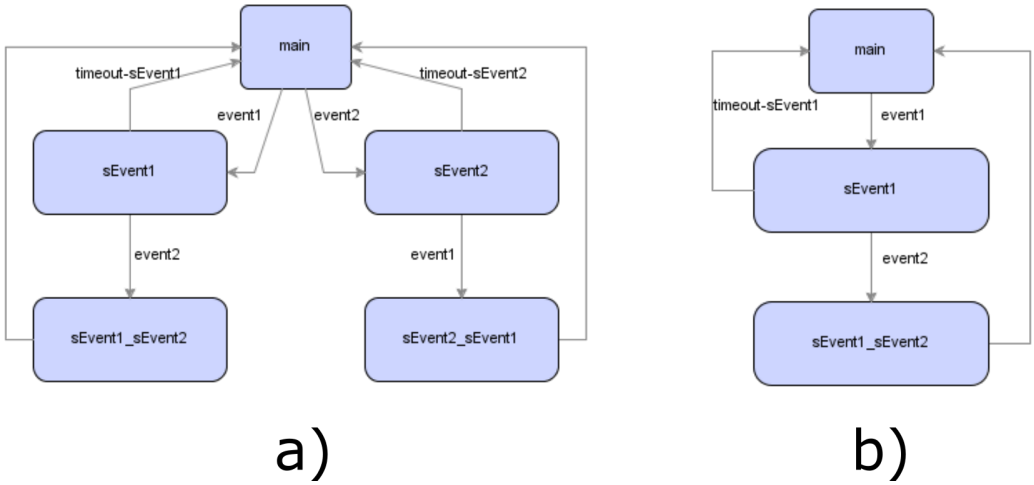

Figure 4-3 – Representation of the SCXML state machine that defines the a) redundancy and b) complementary types

Doing simple steps, it is easy for the developers to create the SCXML state machine to configure the fusion engine. The process is illustrated in Figure 4-4, the developer starts by collecting the Java *enums* generated by each modality and create the correspondent *enum* for the output. The second step is to create a method, instantiating a FusionGenerator object and adding the desired combination by calling the correspondent methods. If two events produce the same output, they can be combined redundantly. For instance, a button to go back or speaking go back. If the sequence of two events produces a new output, the *complementary* method of the FusionGenerator is used. For example, in the context of a news reader application, a button to open the news and speaking how it will be opened (content or image). Finally, the developer only has to execute that code to generate the SCXML.

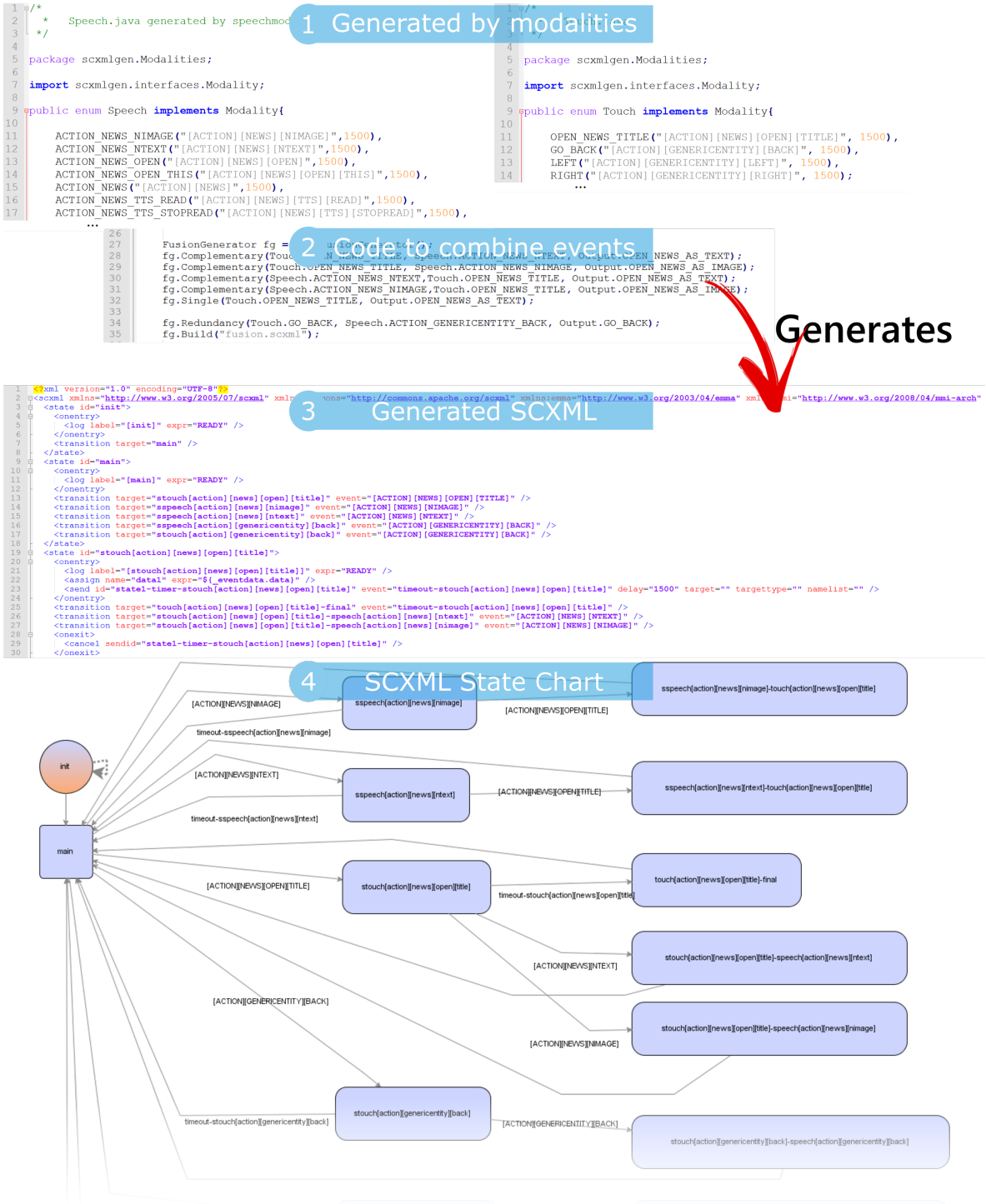

Figure 4-4 –Steps that developers need to do in order to create the SCXML for the fusion engine configuration with a real example. In 1) go to each modality to generate the enum and create the output enum; 2) create the code describing the combination of events; 3) and 4) shows the generated SCXML code and its visualization, respectively.

4.1.3 Runtime

One part of the runtime framework is the definition of the communication and, since HTTP was adopted, the implementation of this is done in each individual module.

The multimodal framework consists of many modules and, therefore, it is unpractical to have to open each module each time it is used. It is the responsibility of the runtime to load and manage all the modules, if for some reason a module stops the runtime loads it again. This way, a part of the runtime is an application that opens all the other modules, shows feedback about the running modules and enables the visualization of the debug messages for each module. Figure 4-5 present a screenshot of the multimodal launcher used in the Paelife[6] project.

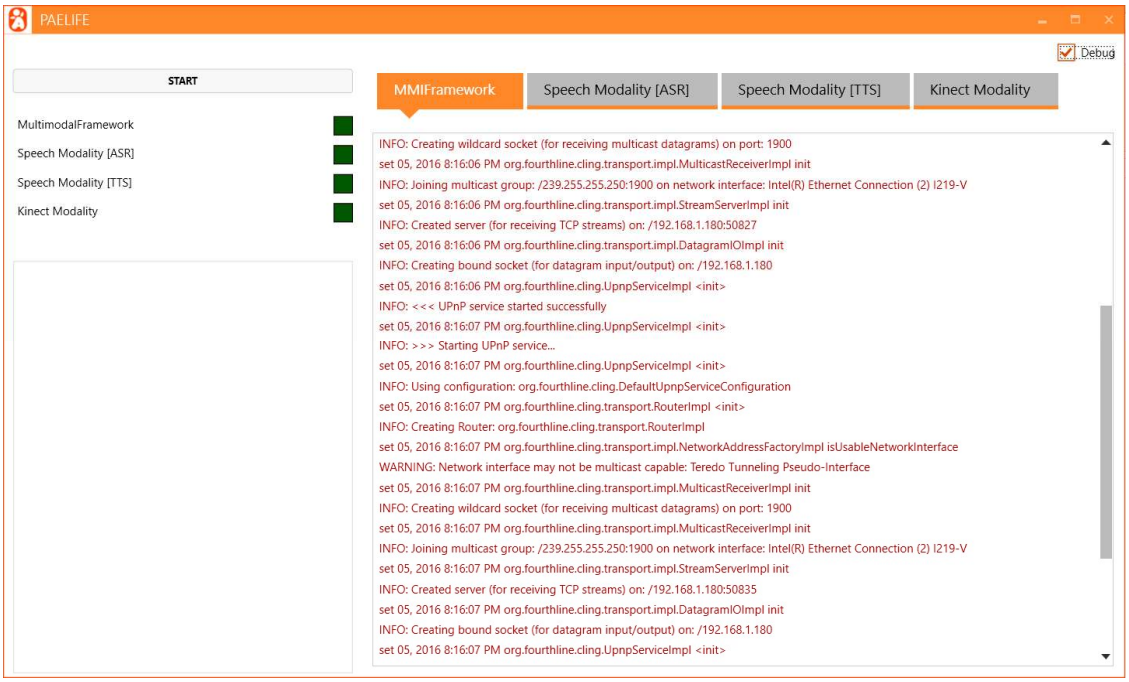

Figure 4-5 – Runtime Launcher of the Multimodal Interaction Framework. The application responsible to start all the necessary modules of the system. It features debug capabilities to help developers understand message exchange.

[6] https://www.microsoft.com/pt-pt/mldc/paelife?lang=en

4.2 Modality Components

Most of the modality components were developed as autonomous applications, so they can be opened by the multimodal launcher built as part of the runtime module. The modalities can output messages to the standard output, so when the developers are testing their applications with multimodal interaction they can analyze the events generated and received. In our approach to the development of the modalities, we have created modalities that are always active and are always read for the interaction with the user.

4.2.1 Generic Speech Modality

As described in the definition of the generic speech modality, the modality was implemented using a number of modules. The developed speech modality supports booth speech input and speech output. Synergies between this work and projects such as PaeLife[7] and Smartphones for Seniors[8] (S4S), enabled creating a more generic modality with more features. Motivated by the context and scope of PaeLife, and the different international partners included in the consortium, it was possible to create and test our vision of a multilingual speech modality and, in both projects, additional voices, also multilingual were added.

Speech Input – The modality, in the part of the speech input, is divided into two parts one running locally and a service[9] running in the cloud. In the cloud a REST-based service allows developers to define a base grammar in English and translate that grammar to other languages (Teixeira et al., 2014; Teixeira, Francisco, et al., 2015). To translate the grammar, all possible sentences that can be extracted from the grammar are extended. Then, using Microsoft Translator Text API[10] all sentences are translated and the grammar is

[7] https://www.microsoft.com/pt-pt/mldc/paelife?lang=en

[8] http://www.smartphones4seniors.org/

[9] The service was manly developed by Pedro Goucha in the scope of his participation in PaeLife. The author of this thesis collaborated in the development and test.

[10] https://www.microsoft.com/en-us/translator/translatorapi.aspx

reassembled maintaining all the original rule names. Since translation is still not perfect, or if the developers want to add other versions for the translations, the service provides a web site to enable manual changes. Figure 4-6 presents the interface of the website.

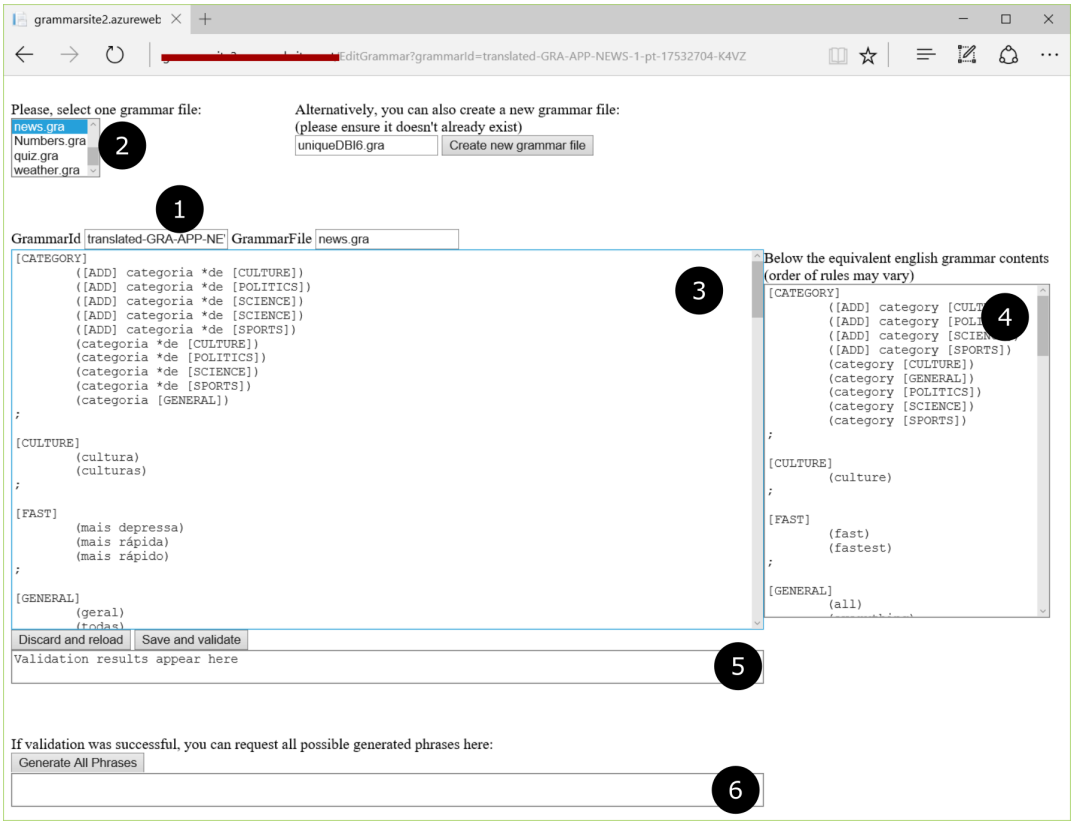

Figure 4-6 - Translation Service management site allowing manual editing of the main and translated languages. 1) Identification of the current grammar; 2) Files composing the current grammar; 3) Automatically translated grammar; 4) Original grammar in English; 5) Area to present the validation results; 6) Area to present all possible sentences.

The service is also used as SLU (Spoken Language Understanding) and it uses the Phoenix parser (Ward, 1991) that, given a recognized sentence and the grammar, can extract the semantic information of that sentence. In order to use the Phoenix parser, the semantic grammar defined by the developer must be in the Phoenix format. Code 4-2 presents an excerpt of an example of the Phoenix format. The syntax of these grammars is very simple. Information is organized by rules, which start with the rule name in brackets and ends with a semicolon. Inside the rules each line is inside parentheses, rules can be called inside these

lines, and the asterisk behind a word or rule means that the word can be in the sentence or not.

```
[Main]
    ([ACTION])
    ([HELP])
;

[ACTION]
    ([AGENDA])
    ([APPOINTMENTS])
[...]
;

[AGENDA]
    (agenda)
    (show my agenda)
    (go to my agenda)
    ([CHANGEDATE])
    (*open [WEEKDAYS])
[...]
;

[CHANGEDATE]
    (change date)
    (select *another date)
;

[...]
```

Code 4-2- Excerpt of a Phoenix grammar specifying a set of rules for the AALFred application

While creating the translation service, an important contribution to this work was the definition of what we called "PaeLife Acts". Based on dialog acts, it provides the semantic information needed to create events. In addition to its

use in the speech modality, it was also adopted by the other modalities. The Phoenix grammar must be defined in a way that, by extracting the name rules, the output is the act. Code 4-3 presents an example of a possible sentence and the result obtained by the service.

```
Recognized: open Friday
Semantic Result: [ACTION].[AGENDA].[WEEKDAYS].[FRIDAY]
```

Code 4-3 - Example of text recognized and the semantic output result

On the client side, the speech input has its speech engine, the Microsoft Speech Platform, offering recognition for several languages, and Microsoft Speech API (SAPI) for additional languages not supported by the Speech Platform. The modality has two operation modes, recognition using a grammar and enabling recognition of sentences directly related to commands or the dictation mode to recognize free text. By default, the modality always starts in command mode and waits for the definition of the language to be used in the current system. When this information is received, the speech modality requests the grammar in the GRXML format for the desired language from the translation service. An simple example grammar, used in an early version of the modality, is presented in Code 4-4.

```
<?xml version="1.0" encoding="UTF-8" ?>
<grammar version="1.0" xml:lang="pt-PT" mode="voice"
    root="Main" xmlns="http://www.w3.org/2001/06/grammar"
    tag-format="semantics/1.0-literals">
    <rule id="Main" scope="public">
      <one-of>
        <item>
          <tag>LEFT</tag>
          <one-of>
            <item>left</item>
```

```
            <item>slide left</item>
          </one-of>
        </item>
        <item>
          <tag>RIGHT</tag>
          <one-of>
            <item>right</item>
            <item>slide right</item>
          </one-of>
        </item>
      </one-of>
    </rule>
</grammar>
```

Code 4-4 - Example of a GRXML grammar

After receiving the grammar, it loads one speech engine depending on the language and loads the grammar to the engine. The interaction manager can request, at any time, a change to dictation mode (if the dictation language model is installed) or back to command mode. When a user interacts and says a sentence, if the modality is in command mode it requests the service to analyze the semantics. Then, the information is encoded in EMMA and a life cycle event is sent to the interaction manager.

The support for new languages, by the speech modality, can be accomplished following a simple procedure. First, the language pack for recognition and synthesis must be installed in the same place as the speech modality. Second, it is only needed to activate the desired language in the service.

Speech Output – the speech synthesis part of the speech modality is continuously polling the interaction manager for requests. When it receives a *StartRequest* event it stops the speech recognizer, so the input does not recognize the output. The message to be synthesized must be encoded as Speech Synthesis Markup Language (SSML), as this markup language enables the configuration of several aspects of the speech synthesis, such as voice, speech volume and rate.

Code 4-5 shows a simple example of SSML that the speech modality can render into synthetic speech.

```
<speak xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               version="1.0"
               xml:lang="pt-PT"
               xsi:schemaLocation="http://www.w3.org/2001/10/synthesis
                http://www.w3.org/TR/speech-synthesis/synthesis.xsd">
    <s rate="-10%">texto para sintetizar</s>
</speak>
```

Code 4-5 - Example of SSML created to synthesize "texto para sintetizar" using an European Portuguese voice.

4.2.2 Body Gestures Modality

The body gestures modality uses the Kinect device to capture the user motion. Using a Kinect framework, developed under the PaeLife project, and the Microsoft Kinect SDK, the framework tracks the user skeleton and, by analysing the skeleton points of both hands, it recognizes the swipe gestures.

When the modality detects a swipe, it sends a life cycle event with the encoded information in EMMA, such as presented in Code 4-6, to the interaction manager. This module is never waiting for requests coming from the interaction manager, so it does not have an HTTP server implemented nor it does polling.

```
<emma:emma xmlns:emma="http://www.w3.org/2003/04/emma"
           emma:Version="1.0">
  <emma:interpretation emma:confidence="1.0"
                       emma:id="kinect-1"
                       emma:medium="gestures"
                       emma:mode="gestures"
                       emma:start="0">
    <command>{"recognized" :
                    ["GENERICENTITY","NAVIGATION","DIRECTION","LEFT"]}
      </command>
  </emma:interpretation>
</emma:emma>
```

Code 4-6 – Sample of an EMMA message generated by the body gestures modality

4.2.3 Touch Modality

Touch, in the context of our scenarios, is very dependent on the graphical interface and the application. For instance, if a button is tapped, the modality has to generate a specific event for the action associated with that button. The proposed multimodal framework provides a library to help with the generation of the events and to handle the communication with the interaction manager. The library contains the necessary methods to send events to the interaction manager, which can be called by the application routines with the desired parameters.

Included in the touch modality is an accessible keyboard, based on statistics, after the user starts writing it finds and highlights the letters that are more probable to come next.

4.2.4 Graphical output

The graphical output (GUI) is a modality, but it is very dependent of the application. In the proposed multimodal framework, it is distributed as an API to be used in the application. The application must start the modality and implement an interface to handle the received messages from the interaction manager. The API already parses the life cycle events and EMMA and provides the application with the event and the necessary data. This way, the developer does not need to have any knowledge regarding the markup languages and communication protocols.

Since this is an output modality, it constantly receives messages from the interaction manager. However, the current approach relies on a polling mechanism based on HTTP GET methods rather than an HTTP server.

4.2.5 Custom modality: Proximity Modality

This modality was developed by Diogo Vieira for project PaeLife and in the context of his Masters Dissertation, to be used in the multi-device scenario, more

specifically when using a first device as a main unit (Home computer with a television) and a portable unit (tablet). This modality must be installed only in the portable unit and connected to a router via wireless. Using the wireless Received Signal Strength Indication (RSSI) it calculates an approximate distance between the tablet and the access point (must be near the main unit). Although this approach is not very accurate and it does not consider walls, it is enough to identify if the user is near the main unit.

Whenever the modality detects a change in the state of the user, near or faraway, it sends the event to the local interaction manager, which informs the other interaction manager. When it sends that the device is near, the two interaction managers exchange events from their modalities, if it is faraway the interaction managers stop communicating.

Figure 4-7 shows the usage scenario of the proximity modality, showing different possible locations of the user and when the user uses the application in both devices.

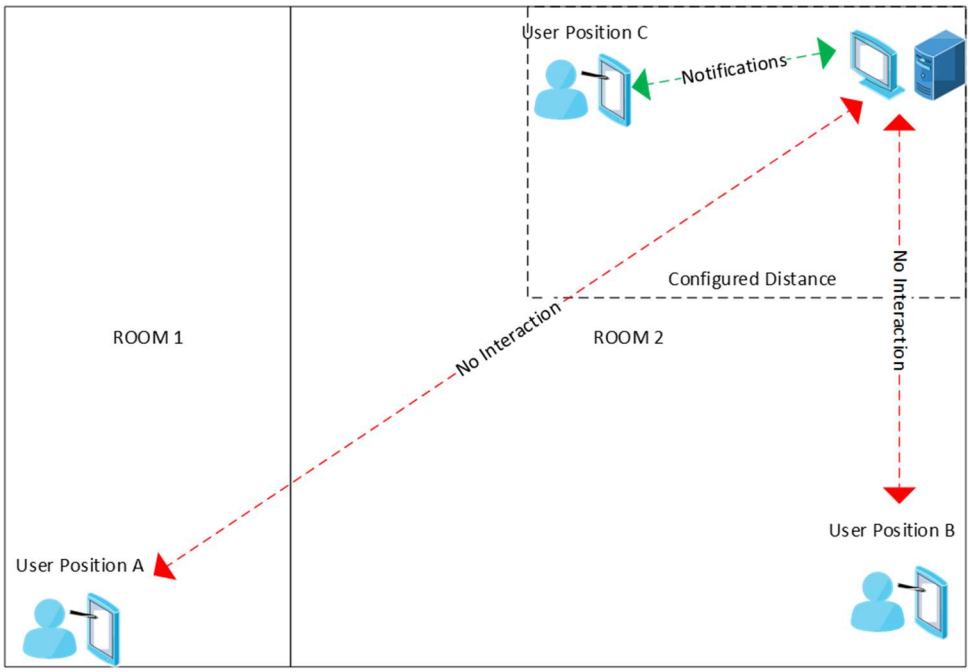

Figure 4-7 - Usage of the proximity modality, the dashed rectangle defines the region where the system considers that the device is near and enables the use of both devices simultaneously.

4.3 Summary

While chapter 3 provided the conceptual and architectural basis for the envisaged multimodal framework, this chapter describes its instantiation, discussing the relevant technical aspects. Details are provided regarding not only the core features of the framework, but also the implementation of the interaction modalities, with a special relevance for the generic speech modality and other relevant features such as multi-device capabilities. However, the proposed framework, as presented in this chapter, still does not fully show its value to address the scenarios and goals we considered at start. In the next chapter, we present a set of applications that, on one hand, serve as proof of the utility and value of the proposed framework and, on the other, worked as workbenches to elicit requirements and promote the continuous improvement of our work.

Chapter 5

Multimodal Interaction Supported by the Multimodal Framework

During the design and development of the multimodal interaction framework, a number of applications were developed, in different contexts, adopting the proposed framework.

One of the first applications, the medication assistant, does not fully adopt the multimodal framework, but allowed to test a multimodal interaction scenario and gather many requirements. As the multimodal interaction framework evolved, new applications were created that adopted the multimodal framework. The first application that fully adopted the framework was a NewsApp, a small application developed in Paelife, in which we could easily test the framework, modalities and the fusion module. But the best example of success of the framework was its adoption in AALFred, started in the PaeLife project and continued in AAL4ALL[11]. The application resulted from the collaboration among development teams from multiple international partners

[11] http://www.aal4all.org/

and, in this context, the adoption of the framework and the proposed speech modality have been a great advantage, facilitating the collaborative work and enabling the rapid development of a multi-language application.

Figure 5-1 depicts a timeline showing the order of development of the main components of the multimodal framework, associated with the involved projects and the applications that have adopted the multimodal framework at that stage.

The projects have provided the context and scenario for the creation of new applications and some of the requirements for the multimodal framework have been determined by the requirements of the applications. The idea of the creation of the multimodal framework emerged in the Living Usability Lab[12] project, where a first version was created with some simple modalities. In this project we have created an application that adopted that framework. In the next project the S4S the requirements forced a different approach in the definition and implementation of the framework. The definition and implementation of the actual framework started with this project and the first version of the speech modality. The medication assistant developed in the context of this project adopts some of the concepts of the framework presented in this work.

The following projects and applications have fully adopted the multimodal framework. In PaeLife we have defined a semantic language common to all modalities to transmit the generated events, evolved the speech modality to be generic and with translation capabilities, and add support for multi-device. The application AALFred, which was started in the PaeLife project and continued in AAL4ALL have benefited from the framework and the already created modalities, enabling the integration of the multimodal interaction in application that was developed by different partners. The module of the news reader created for the AALFred assistant allowed to test the support for multi-device. In AAL4ALL, we had a different approach regarding the support for multi-device applications and tested it using a visualization application.

[12] https://www.microsoft.com/pt-pt/mldc/lul/default.aspx

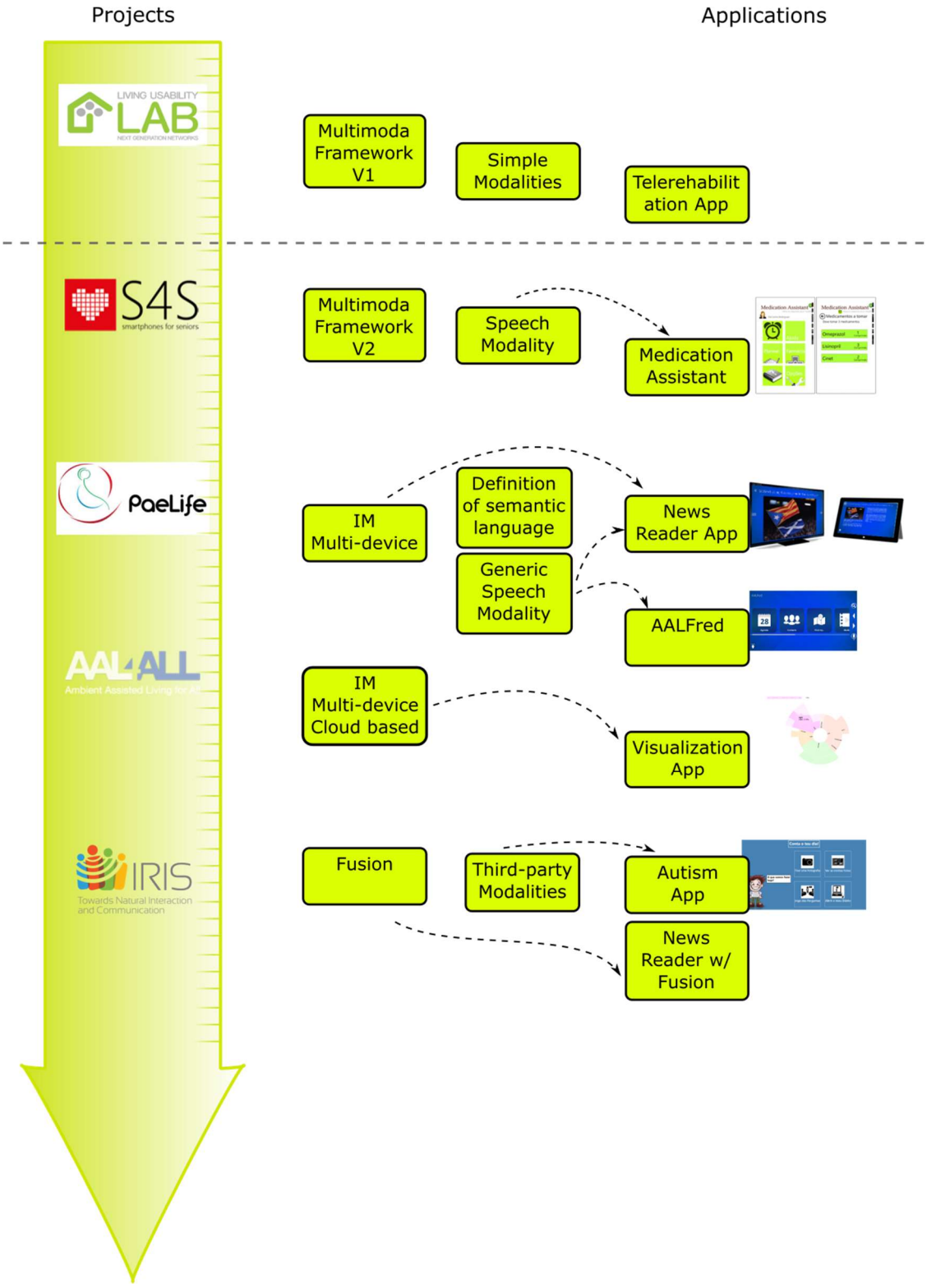

Figure 5-1 – Timeline of the development of the multimodal framework and applications associating the development with the author's involvement in projects.

In project IRIS, a fusion module was developed and tested in a simple application, the news reader application. Also in the context of this project, a third-party has used the multimodal framework, adopting it in the creation of an application for autistic children. Additionally, the developer has created a new modality to use in the application which, given the decoupled nature of the framework, can be used in any other application that adopts it.

In the following sections, the applications that have adopted the multimodal interaction framework are described, highlighting the application context, main requirements and features, and how the proposed multimodal framework contributed for its development. For the sake of simplicity, and given the main purpose of this chapter, some of the aspects inherently involved in designing the applications, such as user characterization (typically through Personas) and context scenarios, are not detailed.

5.1 Medication Assistant

Starting with the fact that the elderly population is growing worldwide (United Nations, Department of Economic and Social Affairs, n.d.) and, associated with that fact, those persons start changing physically, emotionally and cognitively, which impacts the way they live their lives. Helping them to continue living more independently, for a longer time, improves their quality of life and also reduces the costs of health care (Doyle & Bailey, 2014).

New technologies can considerably contribute for improving the daily life's quality of the elderly (Gómez, 2015), making them potential targets for new products, namely application design and development (Doyle & Bailey, 2014). Aligned with European guidelines for the elderly population, which focus in active ageing and preserving independent living, new services based on their specific needs might help them to live in community (Matlabi, Parker, & McKee, 2011).

The non-adherence to medication intake, by the elderly, presents a risk for them and has a negative impact on their lives, by reducing the therapeutic

effects and, inherently, their quality of life. The negative impact can also be felt at the health system, due to the increased cost of having to deal with health situations that might be controlled through medication. Elderly non-adherence to medication is very common in our elderly society and finding new strategies that contribute to improve their motivation to intake medicines, in the right dosage and at the right times is of major importance (Zygmunt, Olfson, Boyer, & Mechanic, 2002).

It is important to engage the target users since the design stage of new products. This way, the products will fulfill more of the users' requirements (Newell, Arnott, Carmichael, & Morgan, 2007). According to (Eisma, 2003) it is even more important to involve users when they are elderly, due to their unique needs and limitations. Working with the target users during the entire process, gathering data on how they use the product, will provide insights of their preferences and guides improvements on the usability of the product (Mynatt & Rogers, 2001).

5.1.1 Requirements

In the beginning of the process to develop the application (Ferreira et al., 2014; Teixeira et al., 2016), many requirements were selected and then grouped as functional requirements and user requirements. The first define the content of the application and the second are more about the interaction.

Functional Requirements

- Medication alerts, to remind users at the time of the medication intake, so they don't forget their medication.
- Medication advice, so users can understand the purpose of a drug, or any side effects that the medicine may have; and, most important, provide advice on how to proceed if the user misses the time to ingest the medicine. Any of those questions must be answered supporting users' voice commands.

- Enable the usage of the application in multiple contexts and the diversity of elderly population by allowing configurations of the multimodal interaction.
- Enable medication management by third parties.
- Record the history of the medication intake, keeping track of the past actions, allowing the user or care takers to view the information.
- The application should run on a mobile device that the user can carry everywhere, for example, a smartphone.

User Requirements

- Allow users to interact in different ways (e.g., touch and speech); offer redundant output and input alternatives since, in some scenarios, users may present a physical or cognitive limitation or just prefer a different modality.
- It is important to be consistent and trustworthy, so users can trust in the alerts and advices given by the application.
- The language that is used by the application should be simple and informal; technical language can be difficult for the users.
- Application should be simple to use and avoid large amounts of content and information at once.
- Allow to adapt the application content based on the users' characteristics and the context.
- Allow the configuration of the application to suit users' preferences.
- Provide help for each functionality of the application in the form of a guide and with short clues while using the application.

In the requirements of the application it is easy to identify some of the main requirements for the multimodal framework, namely the possibility of providing multiple modalities to the users. Predicting its use in other applications, the modalities must be decoupled from the application, so the portability to other application can be simple and efficient. These requirements for the multimodal architecture are also valid to other applications.

5.1.2 Architecture

The development of the medication assistant application begins in an early stage of the development of the multimodal framework. Although many principles of the multimodal architecture are used in the application, it does not use the full architecture. One of the main characteristics of the multimodal architecture to follow was the decoupled nature of the components and its extensibility. In the case of this application, a number of modalities were created, including speech, touch and graphical output. To support the modalities and given the device's limitations, services running on the cloud were created to process data, for instance speech recognition, text-to-speech, natural language generation and others. On the mobile device, those services can be accessed by a network connection (e.g. 3G, LTE, WiFi).

Medication Assistant was developed in a way that application, modalities and services could be updated separately.

Figure 5-2 presents the overall components of the medication assistant application: on the left the components in the phone and, on the right, the cloud services. It is important to note some of the envisaged multimodal architecture features taking form, namely the decoupled nature of the modalities and the evolution of the speech modality to encompass a complex set of features with components running as remote services.

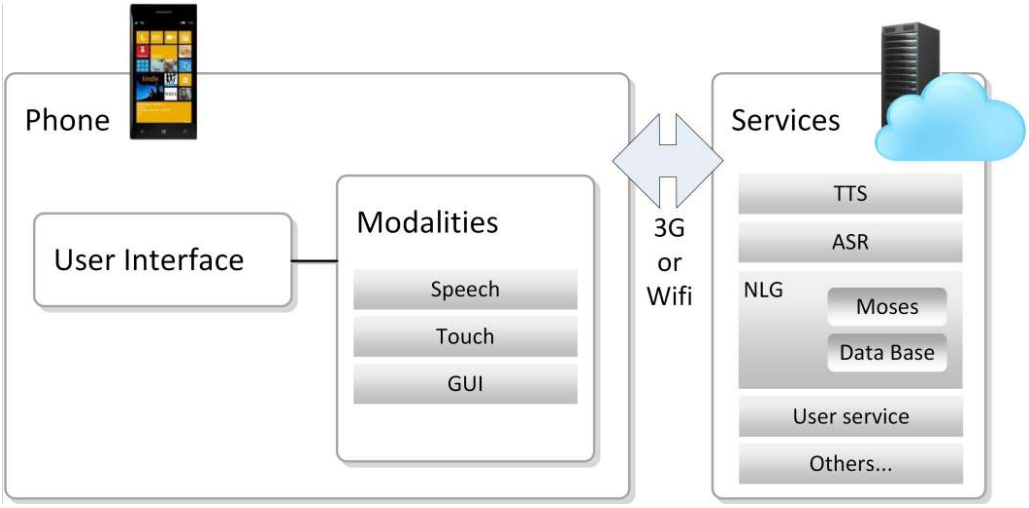

Figure 5-2 - Medication Assistant main components: on the left, the local modules and, on the right, the cloud modules connected through 3G or wifi.

5.1.3 General Presentation

The application implements the selected requirements for a functional medication assistant. Over three iterations, the features presented in the requirements were developed, starting with the most important features (Teixeira et al., 2016). In each iteration, new features were added and the existing ones improved. The medication alarms and the capabilities to provide advices about the intake of the medications were considered the most important requirements to fulfil. Support for multimodal interaction was also considered as a priority, aiming to provide a more intuitive interaction between user and application. Other features were added in later iterations.

Figure 5-3 shows screenshots of the application after the first iteration: (a) depicts the main menu of the application enabling the user to select a functionality of the application; (b) presents the list of current medication and posology; (c) shows a menu that allows users to ask for advice if they forget to take a medicine or if they are felling side effects; finally, (d) if a user chooses to select any option using voice commands, and the system does not recognize the command, a message appears containing information about the recognition problem and providing suggestions for possible voice commands.

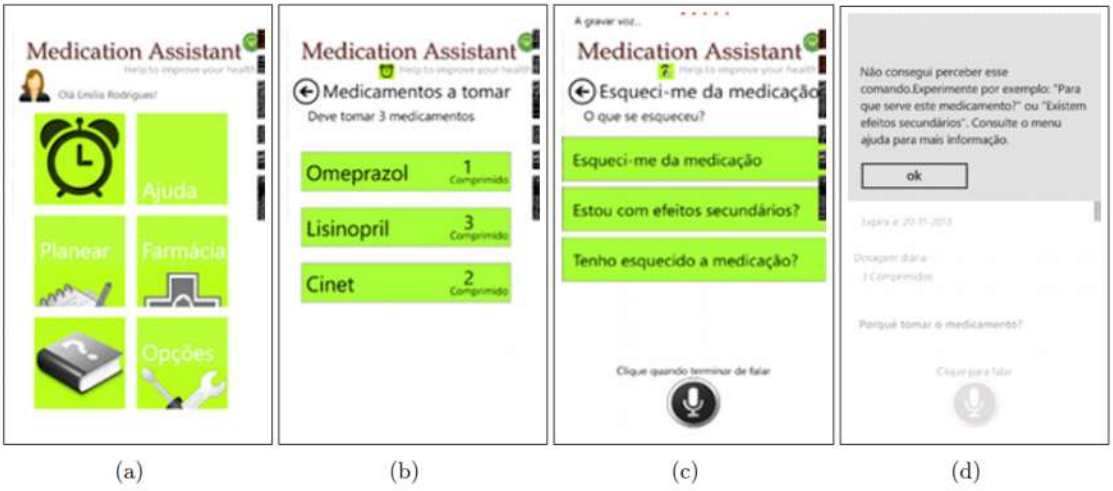

Figure 5-3 - Screenshots of the medication assistant (1st version). (a) main screen; (b) list of prescribed medicines; (c) advices when the user forgets to take the medication; (d) message suggesting voice commands when the system did not recognize a command.

This first iteration counts with two parts, alarms that contains the list of drugs and advice with relevant information of the medicines. In the following iteration other requirements were instantiated in the application, the insertion and management of the medicine plan, new views with information about each medicine were added, capable of showing an image of the pill based on a description. This views also are adaptable, image size can be zoomed in or out according to the distance of the display to the users eyes. Important to note, users can use the application with the usually used modality in smartphones (touch and display images and text), but also user speech input and output.

5.1.4 Interaction Implementation

At the beginning of the development of the medication assistant application three modalities were considered: graphical user interface (GUI), speech and touch. The importance of using the decoupled architecture is highlighted by the fact that modalities can be developed separately and later improved or exchanged by new modalities. This way, developers of the application focus on the application and developers of modalities only focus on modalities. Also, modalities can be reused in other projects.

Regarding the speech modality, previous work (Almeida, Silva, & Teixeira, 2014a) was used to instantiate the speech modality including speech recognition (ASR) and speech synthesis (TTS). A part of the modality is running on the used device and the other part is running on a cloud service. For speech recognition, the sound is recorded, in the device, and it communicates with the service to obtain the result. The service must be configured using GRXML grammars (Hunt & McGlashan, 2014) and a speech recognition model, which can be changed at any time.

The speech synthesis works the same way: the service converts the text into speech and the modality in the device plays the audio stream. The service enables choosing from a set of five different voices, so the user can choose according to his preferences. Since the speech engine considered was the Microsoft Speech Platform, it natively provides one female voice available. The other four voices (two older adults and two young adults) were created under the S4S project (Almeida et al., 2015) and are an important addiction to enable a greater versatility and adaptation to user preferences and context.

A Natural Language Generator (NLG) service was integrated with the speech modality and it complements the speech synthesis. Instead of having a limited number of sentences to be read to the user, the NLG can generate new sentences based on the information to be transmitted. The NLG uses a statistical machine translation system configured with a trained language model that can be adapted to fulfill the requirements (J. C. Pereira, Teixeira, & Pinto, 2012). The following example illustrated the functionality of the NLG: given an input with several arguments it returns a sentence that has sense.

Input: 2 7 2 160 8
Output: "Do not forget to take two green pills of drug GGG every 8 hours."

While the modality was created to support the European Portuguese language, it is easy to extend it to support other languages. Developing a speech modality may be a difficult and time consuming task, but by using this methodology the modality can be easily reused in other works.

In order to improve the visual content of the application, a service to generate images of a pill was also created. Given an input with the description of the pill, such as shape, colors and text it generates a synthetic image of the pill.

Other services, not related with the interaction, were created to manage the application information. For example, the user service manages the user data including the medication plan.

5.1.5 Evaluation and Results

Since the application was developed over a number of iterations, an evaluation was performed[13] in each iteration, allowing making improvements to the first requirements. A methodology of evaluation based on the ICF-US US (International Classification of Functioning based Usability Scale) (A. Martins, Queirós, & Cerqueira, 2012) was used. It consisted in the users testing the application and in the end answering a questionnaire to assess the application's usability.

Different aspects of the application were evaluated – ten in total – and classified as a barrier or facilitator (from -3 to 3). The sum of all items gives the usability score. A score near 30 entails that the product has a good usability.

The Figure 5-4 (a) shows the score of each user and Figure 5-4 (b) the average score of each item. It is worth noting that, using the described scales, the classification can be negative (with a minimum of -30), but the obtained scores are well above zero. The results obtained in the evaluation were all positive, with a minimum score of 19 and a maximum of 26. The average score, among

[13] Collaborative work in the scope of the S4S project, in which the author participated.

all users, was 22. In (b) it is visible that the items "status feedback" and "ease of learning" were the ones with lower scores, 1.8 and 1.7, respectively.

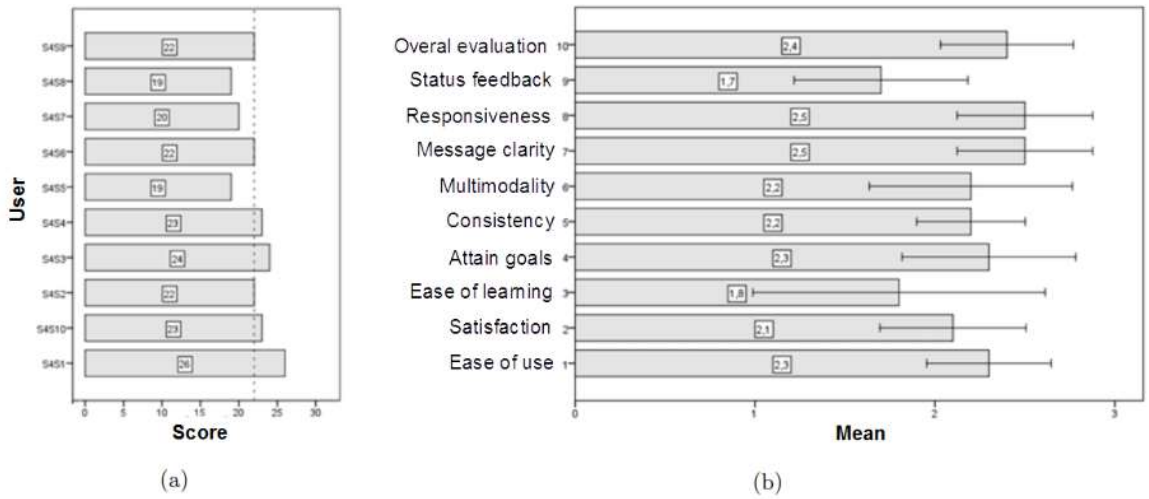

Figure 5-4 - Questionnaire results: (a) score for the application given by each user; (b) average score for each item (question).

Also, other difficulties were registered while the users worked with the application. In Figure 5-5 it is easy to see that the users who performed the evaluation found the swipe gesture a difficult task.

Figure 5-5 - Difficulties using the medication assistant: the larger scores correspond to increasing difficulty.

5.1.6 Overall Remarks

Despite the application's use of an early version of the framework, it allowed a greater insight on how users interact using a multimodal application, if they find it useful and their main difficulties. This application was also important to support the development of the first version of the generic speech modality.

5.2 AALFred – Personal Assistant

Worldwide, the number of people aged over 60 is constantly growing (Organization, 2002). Noting an increasing change in their social lives, the World Health Organization is making efforts to keep this generation active, productive and independent by promoting policies and support programs.

Applications that foster social integration and participation, and enable safer day-to-day independent living, for elderly persons, could translate into an active ageing, inserted into the community and fighting isolation. In this context, in project PaeLife a personal assistant named AALFred was proposed.

5.2.1 Requirements

The initial requirements for the envisaged application focusing seniors were obtained through brainstorm:

- Support for different modalities, with a particular focus on speech, since it is a natural form of communication that can be potentially easier to use by older adults;
- Create a multilingual application, (since the application started as a part of an European project – PaeLife) supporting at least the native languages of each partner of the project;
- Have a decoupled solution, to allow distribution of modalities across different devices and to support better integration of modules done by partners;

- Create modules to support different social services and other information services;
- Creation of an application hub to enable the integration of the different modules

As happened with the previously presented application – Medication Assistant – the application requirements include the framework requirements. The decoupled architecture is now particularly important in this development scenario, with multiple partners working in one application. The definition of a unified semantic language for the communication of the modalities and application should enable developers to only focus on the application.

5.2.2 Architecture

Figure 5-6 shows the architecture for AALFred. It depicts, on the left side, the devices inside the user's home, which connect to the cloud services, represented on the right side. The AALFred application fully adopted the proposed multimodal framework. So, events related with interaction are managed by the developed interaction manager. The framework also integrates many modalities, including the speech modality and gestures.

The graphical user interface modality is a part of the application and it is connected with the interaction manager to receive new messages. The modality is responsible for calling the corresponding methods in order to update the displayed information in the application interface.

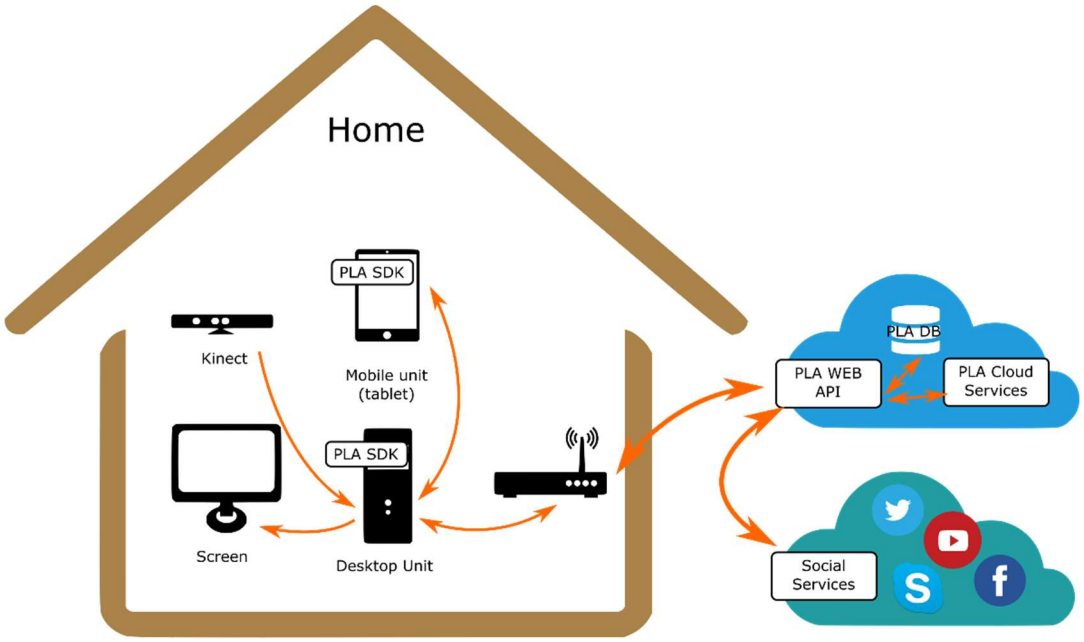

Figure 5-6 - AALFred Architecture: inside the house, the users' infrastructure and applications supported with the Personal Life Assistant (PLA) SDK; on the cloud, the PLA services that provide support and content for the applications.

An overview of the connected components is shown in Figure 5-7.

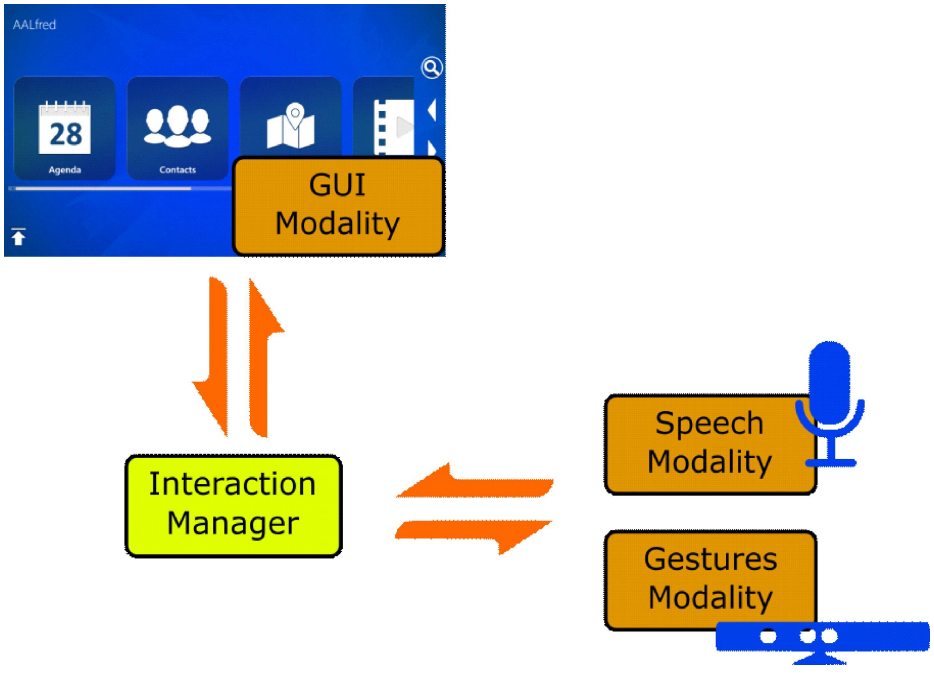

Figure 5-7 -Representation of the main modules and their connections while using the personal life assistant AALFred. The GUI modality is coupled with the application and connects to the interaction manager. Speech and gestures interaction can be used through the available decoupled modalities.

The interaction manager controls the flow of events related to interactions. Every time an events occurs in speech, touch or gestures modality, the modality sends a life cycle event notification to the interaction manager, the event is processed in the interaction manager and an event is sent to the graphical user interface modality, in order to change what is shown in the interface. If the application has something to be read to the user using speech synthesis, it sends an event to the interaction manager, which is responsible for delivering the event to the speech synthesis modality.

Any of the available modalities allows the user to interact with the application, often redundantly. For instance, to slide the container with the different modules, any modality of the input modalities can be used:

- With Touch we can drag the content;
- Via Kinect it is possible to swipe a hand to the left or right;
- Speech simply allows for actions to be active via words such as “left” or “right”.

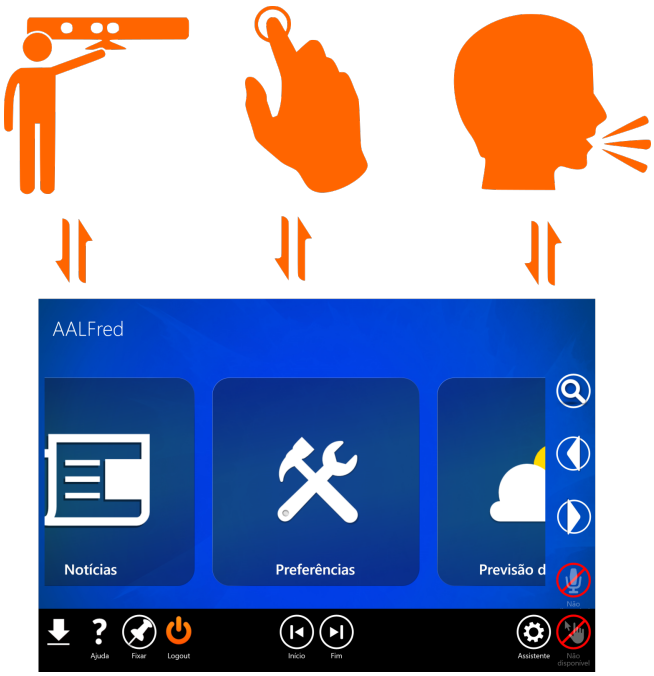

Figure 5-8 - Presentation of the ways to interact with the AALFred application. From left to right: gestures, touch and speech.

5.2.3 General presentation

AALFred, is a personal assistant application containing a collection of services, information services and social services, which was developed in collaboration of multiple partners. The application integrates the developed multimodal framework, users can interact with the application using any available modality. Each feature of the application can be accessed for instance with touch or speech. Figure 5-9 present the initial view of the application, the menu used to navigate for any of the modules developed for the assistant. The modules include: Agenda, Contacts, Messaging, Media, Find My…, News Reader and WeatherForAll.

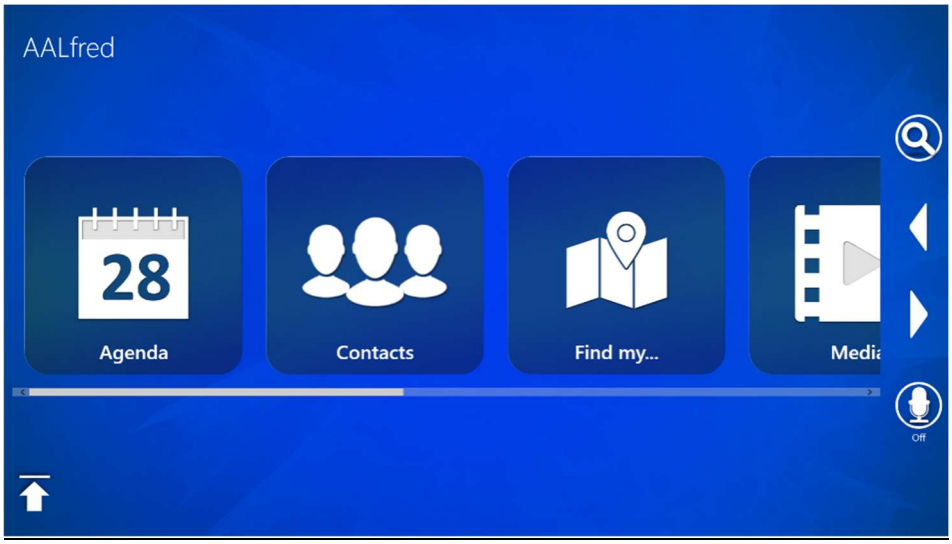

Figure 5-9 - The initial screen of the application showing the list of modules.

An example of usage of the agenda is given in Figure 5-10, after open the agenda the user can choose to use touch or speech to navigate in the module according to his preference.

The AALFred assistant, while user navigates into new modules gives suggestions of the features available and what the user can do. This information is transmitted to the user using speech synthesis, for instance entering the agenda: “Do you want to create a new appointment? Open a day and say create a new appointment”.

Figure 5-10 - Interactions Flow to create a new appointment in the agenda. The user touches the Thursday button or speaks "Thursday" to open the appointments for that day. To add a new appointment uses speech or touch. Finally uses the accessible keyboard to write a text.

5.2.4 Implementation

The application itself is divided in several modules, which were developed by different partners of the PaeLife project. In the end, the modules were integrated in the hub application. Each module adopts the proposed concepts and methods to develop multimodal applications using the proposed multimodal framework.

The modules were developed including the support for string localization, so each text in the application appears in the language of the user, including the sentences that are synthesized and read to the user. In the case of the speech recognition, the speech modality uses a developed service, which allows the automatic translation of grammars (Teixeira et al., 2014), as previously

described in this document. This way, the speech modality generates the same output for sentences in different languages that have the same meaning. Also important to note is that the output is unified among modalities. So if the user taps the button to open or speaks “my agenda” or “open agenda” the output of the modality reaching the interaction manager will be the same in each case. This, not only simplifies development, but also eases the configuration of the interaction manager’s state machine. Any of those user actions will result in the application opening the module “Agenda”, as presented in Figure 5-11.

Figure 5-11 - Screen of the application, with the agenda module presenting the days os the current week.

5.2.5 Tests and Results

The evaluation of the AALFred personal life assistant was conducted by the involved partners of the project. Microsoft and the Portuguese national guard (GNR – Guarda Nacional Republicana) have performed a large scale pilot test, for the Portuguese version and the results show that the system has provided an overall positive experience for its users. A more detailed description of the

results can be found in the AAL4ALL final Technical Report (AAL4ALL-Consortium, 2015).

5.2.6 Overall Remarks

The decoupled nature of the architecture allowed the development of the modules without having to be concerned with the complexity of the modalities. At the end, by following the methodology it was possible to use different modalities, even modalities created after the application. Another advantage resulting from the decoupled nature of the framework was the successful development of an application by different international teams, developing in parallel and without any need to attend to modality specific issues in their modules. Also, the framework simplified the work of using different languages, concerning the speech modality, due to the automatic translation of grammars.

5.3 Multi-device News Reader

The use of mobile devices such as tablets and smartphones is widespread and many persons have at least one of those devices while keeping their fixed computers, which can be connected to a large screen. Also, applications can run seamlessly on different devices, only by adapting the content to the screen size.

Since users can run the same application in different devices, bringing the two applications to work together could enhance the usability and deliver more information at once to the user by taking advantage of the characteristics of the different devices.

5.3.1 Requirements

The requirements for this application reflect the ones defined for the AALFred application, particularly regarding the support for multimodal interaction, including speech, touch and gestures. Additionally, new

requirements were determined to create a multi-device experience. The application should:

- Be capable of running independently on any device.
- Enable the connection of one application running in two different devices.
- Detect proximity to start working together and detect when they are distant to work independently.
- Be capable of showing the same content in both devices, or different content for the same subject.

The support for a multi-device application is managed by the multimodal framework and it was in the context of this application that novel requirements were set for the multimodal framework. The interaction manager should be capable to connect to another instance of the interaction manager and it should adopt mechanisms to discover and register the existence of the other interaction manager for future communications. The support for these features must be independent of the number of devices and users can choose to work with one device or two, depending on their context.

All the requirements defined in the previously presented applications are applied in this update of the multimodal framework.

5.3.2 Architecture

Since one requirement for the application is the possibility of running the application in each device separately, each device now runs an instance of the interaction manager. It is mandatory that each device runs the application and one interaction manager. Modalities can run on any device and communicate with the interaction manager that is running in that device. Figure 5-12 shows the architecture of the multi-device, showing that devices only communicate through the interaction manager.

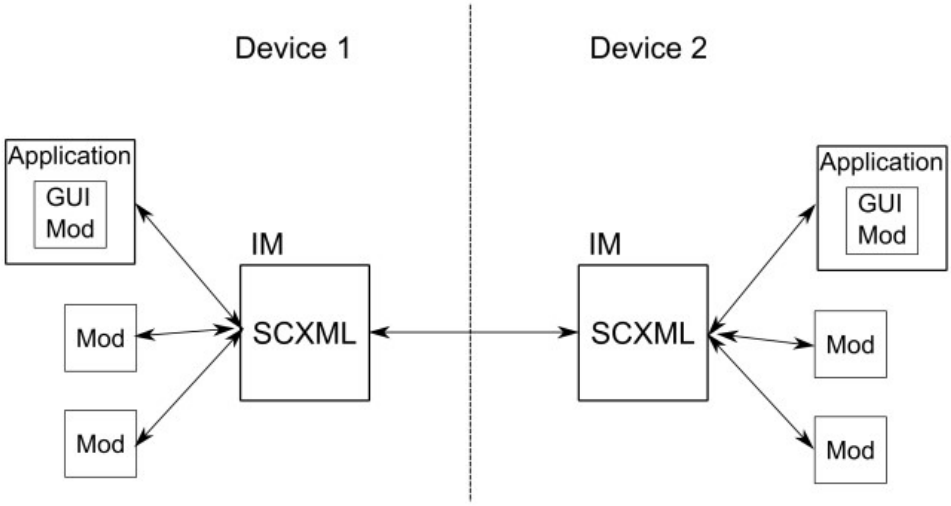

Figure 5-12 - Multi-device multimodal architecture

5.3.3 General presentation

This application is an update of the news module developed for the AALFred assistant, initially designed for being used in a single device. After this update, the news content can now be shown to the user in different modes.

Basically, news content can be displayed in three different ways: as a small image with the news text, showing only a large size version of the image, on the screen, or as a list of news in a tiled view. Using these different views it is possible to make combinations considering the simultaneous use of two devices. Figure 5-13 presents some screenshots of the application illustrating possible combinations considering a large TV set connected to a computer and a tablet:

- TV showing the content of the news / tablet also showing the content
- TV showing the image illustrating the news in full screen/ table showing the content with the text description
- TV showing the content / tablet continue showing the list of news, so the user do not have to go back to select other news.

Figure 5-13 - Multi-device scenario, showing multiple ways to present information when two devices are available. From top to bottom: both devices presenting the same content; one device presents a full picture of the news and the other all the content; one device presents the full content and the other the list of news.

Naturally, other combinations might be accomplished using the application, but these were the ones considered relevant for the use context. If, at any point, one of the applications is set to display only the news list and it starts working alone (e.g., by moving away from the other device), the application reverts to working autonomously.

5.3.4 Implementation

For the multi-device news reader, most of the development work was inherited from the previous application with the addition of a new view showing the images and the possibility to change the mode of information display. The interaction manager was updated to support the discovery of other interaction manager in the same network. To accomplish the discovery of new devices running the application and one interaction manager the uPnP protocol was used. If a device enters a new local network it sends a broadcast message to devices, if the other device receives this message, it registers a new interaction manager and its address for later use. It responds to the broadcast message with its address so the other interaction manager can also register its existence. Also, the interaction manager can now send extension notification life cycle events for others interaction managers, reproducing received events in the other manager.

Figure 5-14 shows the flow of life cycles events, after the two interaction managers discovered the existence of the other, and the messages exchanged when events occur.

It was in the context of this application that the new proximity modality, described earlier, was created. Usually running on the mobile device, and assuming that the TV is near the wireless router, it calculates the wireless signal strength allowing to sense if the device is near or far from the TV. If it changes from near to distant or vice versa an event is triggered, resulting in changes between single-device and multi-device functioning.

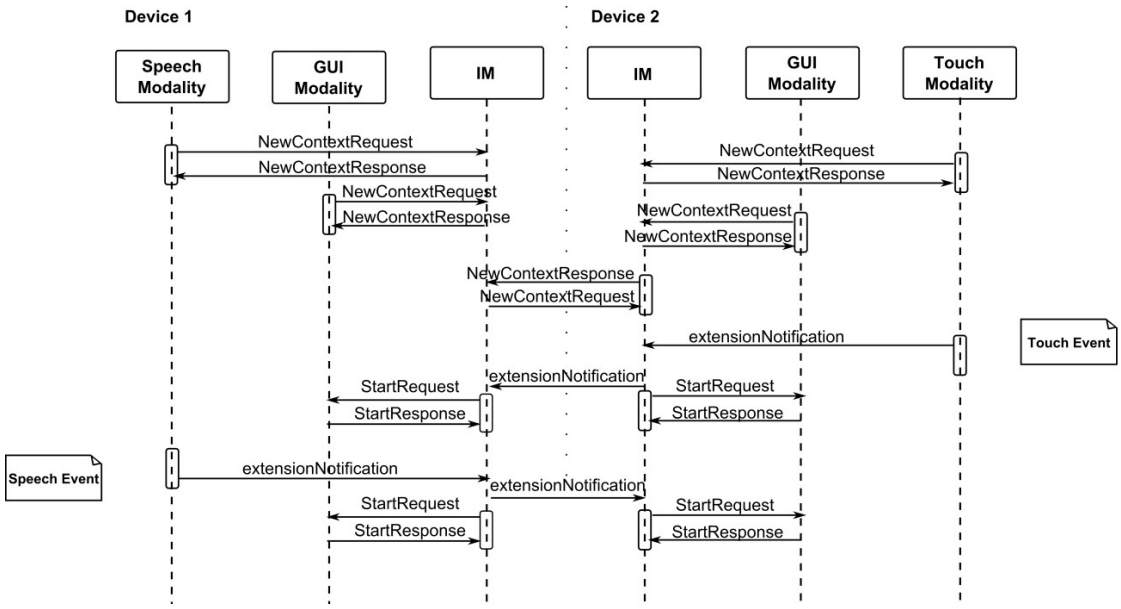

Figure 5-14 - Flow of life cycle events in a multi-device scenario

5.3.5 Tests and Results

The news reader application is also a module of the AALFred application and was tested and evaluated in that context. The usage of the news reader in a multi-device scenario was only informally assessed by the author.

5.3.6 Overall Remarks

With a small effort, it was possible to use the multimodal framework to create a multi-device and multimodal application. The application running in two different devices can work together, delivering more content to the users while keeping the ability to run separately, as usual.

One of the notable points highlighted by this application is how the decoupled nature of the framework allowed adding the multi-device feature to an existing application with only minor changes to the application core.

5.4 Multi-device Visualization Application

Dynamic Evaluation as a Service – DynEaaS (C. Pereira et al., 2015) – is a platform that supports the evaluation of complex multimodal distributed systems. The platform collects all data concerning the users' interaction with a system, organized in a hierarchical form, according to the application components. Some insights of the users' performance with the application can be extracted by analyzing this data. This way it is important to create visualizations of that data and allow experts to interact and discuss it.

5.4.1 Requirements

Given the main objectives considered for the visualization application, namely to present information and enable the interaction with different visualizations, a few requirements were defined:

- Create different visualizations, showing the same data but in a different way;
- Support multiple devices, adapting visualization to the screen size;
- Support collaborative visualizations.

Most of the framework's initial requirements are already covered and the framework already supports much of the required features. In this scenario, a new GUI modality is required, which must have adaptation capabilities. The way and the amount of information that will be presented should adapt according to the display capabilities.

The requirement to support multi-device also applies to this application. However, since a more generic scenario is envisaged, including an unlimited number of devices, and given possible limitations of some devices, a different approach had to be taken. Therefore, this was the application where a novel approach considering a centralized interaction managed was first tested, managing all interactions from all devices.

5.4.2 Architecture

The visualization system adopts the multimodal framework and in this approach only one interaction manager is used and is responsible for managing all the life cycles events coming from all devices. All visualizations in one device are managed as only one modality, the touch is part of the application, and other modalities are allowed to connect to the framework.

Other modality was created to use the smartphone capabilities, using the accelerometer to detect the smartphone motion. The user can rotate the smartphone 90 degrees to the right or left to navigate through the data.

Figure 5-15 depicts the overall approach, with the centralized interaction manager and possible devices, showing a visualization of the data.

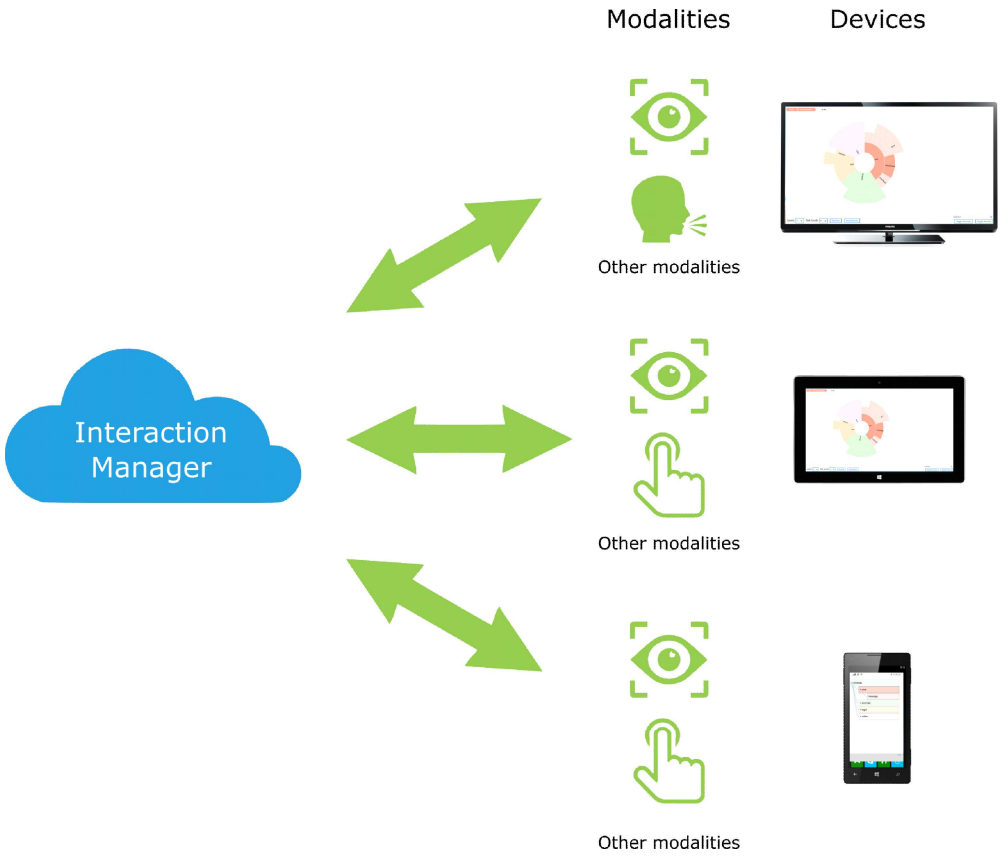

Figure 5-15 – Multi-device approach for the visualization application. Different kinds of devices run the available modalities, which are connected to a central interaction manager.

5.4.3 General presentation

To support the design of the application, a scenario of usage was considered, describing a meeting between three experts discussing the results of an evaluation session. The meeting room is equipped with a computer connected to a large screen running the application and users bring their own devices to the meeting (tablets, smartphones, etc...) and run the application in the device. While discussing the data, when a user interacts with the application all the other devices show the result of that interaction. All visualizations in all devices are synchronized and all devices react to the interactions made by any member of the group.

Users are able to select from four different visualizations (sunburst, treemap, treeview, timeline view), allowing them to select the best suited visualization for the data (and device) or the one that they understand better. The personal choice of each user does not influence how the remaining users see the data. Adding to the proposed visualizations, other techniques were made available to help navigate and visualize data (tooltips, breadcrumb).

5.4.4 Implementation

Following the choice made for previous works, the application supports devices with Windows, either desktop, tablets and smartphones. To implement the visualizations the D3js[14] framework was used. The following figures shows all the possible visualizations. Figure 5-16 shows the sunburst visualization with breadcrumb and tooltips, Figure 5-17 the treemap, Figure 5-18 the treeview and Figure 5-19 the timeline with tooltip.

[14] https://d3js.org/

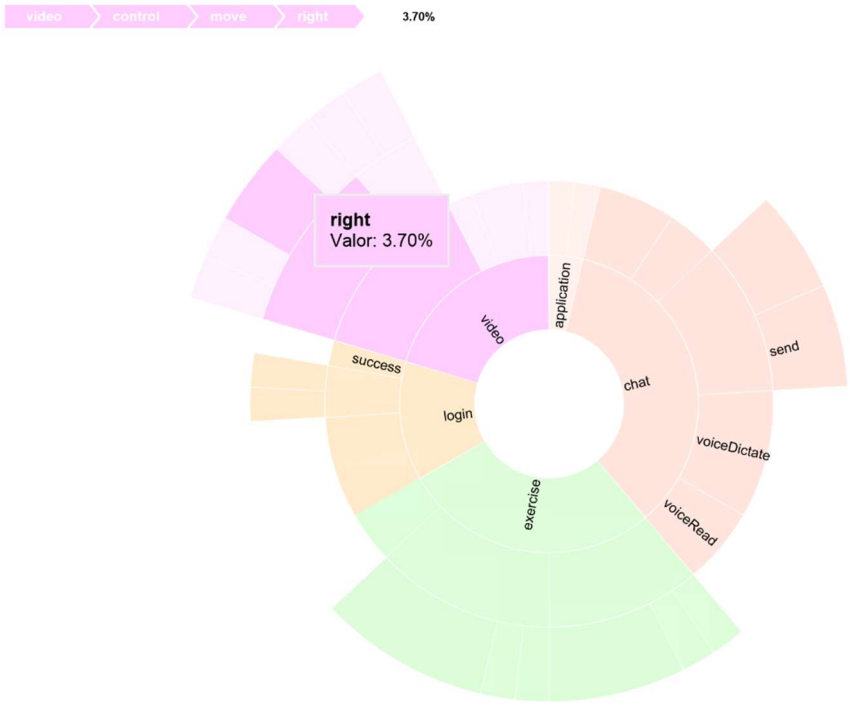

Figure 5-16 - Example of Sunburst visualization available in the multi-device visualization application

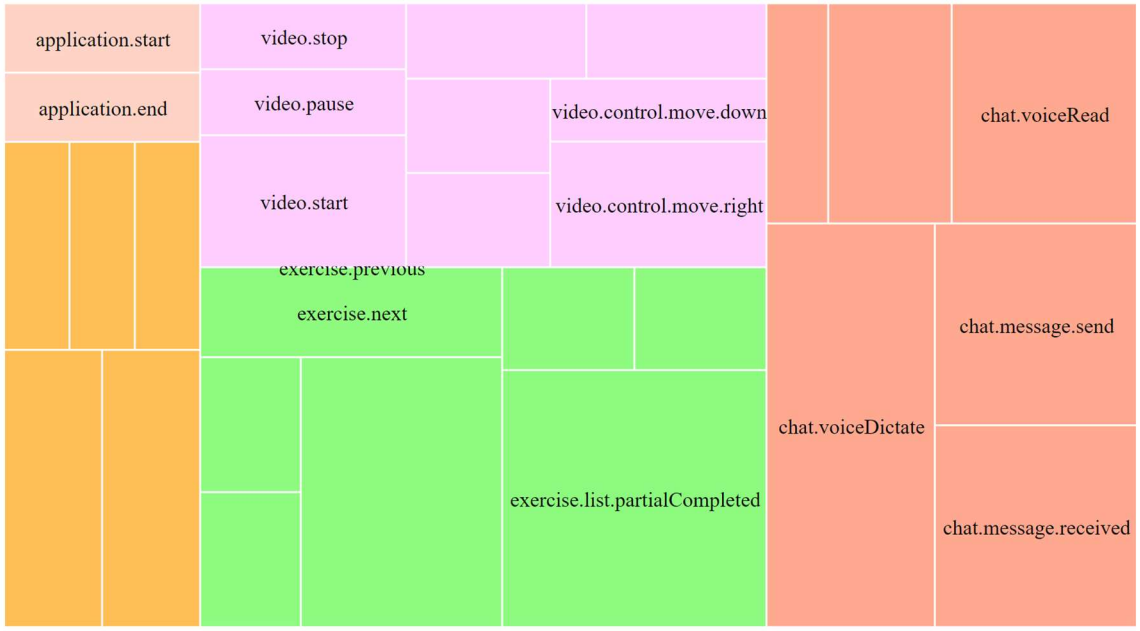

Figure 5-17 - Example of Treemap visualization available in the multi-device visualization application

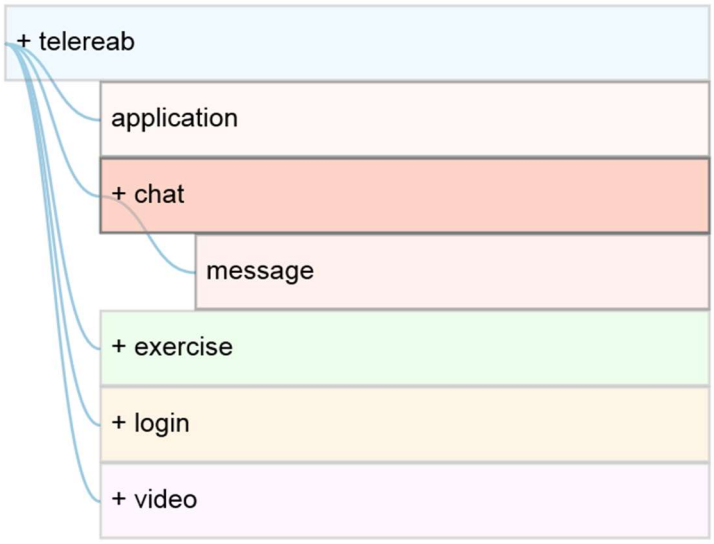

Figure 5-18 - Example of Treeview visualization available in the multi-device visualization application

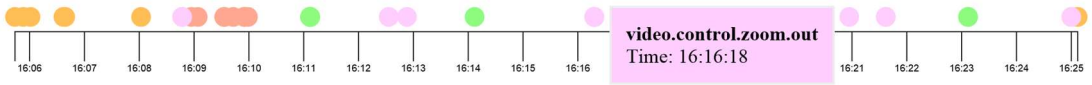

Figure 5-19 - Example of Timeline visualization available in the multi-device visualization application

5.4.5 Tests and Results

To assess the application performance, a plan based on (Pinelle, Gutwin, & Greenberg, 2003) was created to evaluate it when used by single users and in group. Users were asked to complete two sets of tasks (see Table I), the first set of tasks to be conducted individually, and the second set in group, with each user working on a different device (PC, tablet or smartphone). In the second set each user had a personal task, but others might also interact to help finding the result faster.

A subjective approach was used. Users were observed performing the tasks, a log of incidents was registered and users were encouraged to think aloud. In the end, users were asked to fill a questionnaire, based on the System Usability

Scale (SUS) (Lewis & Sauro, 2009). Furthermore, using the same scale as the SUS, other items were added to the questionnaire to analyze user preferences regarding the visualizations and their usage in multi-device conditions.

Although the scores obtained using the SUS were low, users found it helpful having different visualizations and the way the smartphone could be used.

5.4.6 Overall Remarks

The support for multi-device in the visualization application had a different approach. It uses only one interaction manager to tackle all running applications in the different devices and all modalities. It is easy to add new devices running the application or only running modalities.

While the first evaluation results were low, this entails a first instantiation of a novel paradigm where multi-device capabilities, enabling collaborative efforts, is intrinsically provided by the architecture, rather than as a specific application running on a conference room. This means that such features are made available, from the start, to all applications that adopt the framework. Therefore, we believe that these features have a strong potential and deserve additional attention

5.5 Application for Special Needs

It is not new that information and communication technologies can improve several user tasks. In the context of users with special needs, more specifically students with autism, the use of such technologies can contribute to improve the learning process (Liu, Cornish, & Clegg, 2007; Williams, Jamali, & Nicholas, 2006). This application (Vieira, 2015) is an example of an application developed by others using the proposed framework.

5.5.1 Requirements

This application targets a special group of users, and its design and development must be careful to consider relevant aspects of their characteristics. Through a brainstorm session, including people with different backgrounds, some directly related with autistic children (e.g., speech therapist) some requirements were determined:

- The application must provide features for two different types of users: the autist child and his teacher.
- Provide functionality for the creation of quizzes and for the child to play them.
- Allow the child to maintain a diary, with the possibility to share some of the content of the diary with family and friends.
- Enable the user to take photos and manage them.
- Integrate the speech modality and have a new modality that enables interacting using gaze.

At this stage, the multimodal framework had already reached most of the goals initially established and fusion of events was the last of the framework requirements missing. The framework must have a module to combine events from different modalities.

In this context a modality for eye tracking was necessary, and this application provided the context where a third party developed and integrated a novel modality.

5.5.2 Architecture

The application provides two features, the diary and a quiz. Figure 5-20 shows an overview of the application modules. The quiz set up in the by the teacher in the application opened in other table, the two application

communicates with each other. The diary besides storing the child's photo allows publish that content to a social network.

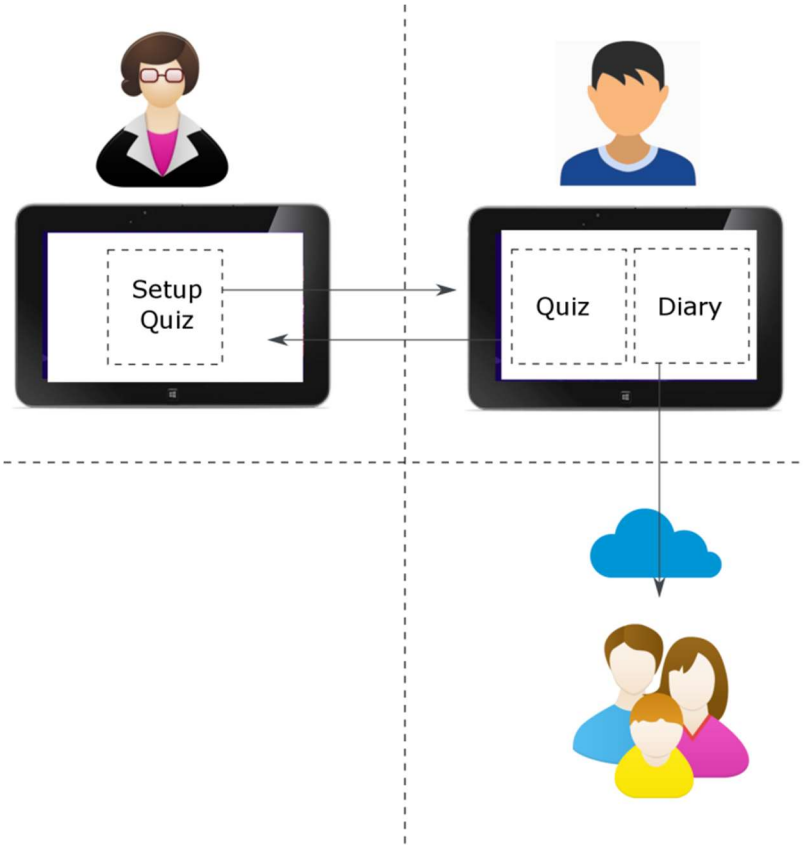

Figure 5-20 - Overview of the main modules of the application, distributing each module for the target users.

5.5.3 General Presentation

The application is divided in two sections, one to serve the child and other for the teacher. In the child section, it enables the child to take pictures, save those pictures in the device. Also the application allows the child to publish the photo along with a comment and an emotion, in a diary and share with is family and friends. Still, in the child section the application provides an interactive quiz, which is set up in the teachers' section.

The modalities in the context of this application allows the child to interact with the application by speech, touch or eye gaze, interaction with modalities

can be used independently or by combining two modalities. Figure 5-21 present the initial view of the application used by the child, allowing him to navigate to the features of the application.

Figure 5-21 - Screenshot of the autist app

5.5.4 Implementation

Following previous works, the application has adopted the multimodal framework, and uses the speech input/output modality and touch. In addition, a new modality was developed, the eye gaze modality, which allows the detection where the user is looking at, on the screen. This implementation, by a third party developer, proved that it was easy to add new modalities to the framework and continue to use the others.

With some changes to the state machine, part of the interaction manager, was possible to make fusion between speech and eye gaze events and this was the first evolution towards the fusion module later proposed by the author.

The connection between the two applications in each device uses a discovery protocol.

5.5.5 Tests and Results

The application was subjected to usability tests and the evaluation, based on the ICF-US US (International Classification of Functioning based Usability Scale) (A. Martins et al., 2012) scored 17.7, which indicates that the evaluated prototype has good usability. During the evaluation some enhancements which must be done were determined.

5.5.6 Overall Remarks

Apart of the results of the evaluation, which were good, a new modality was created and used in this application. The development work of this modality was done by a third part developer, which seamlessly integrated the new modality with the multimodal framework.

5.6 Summary

After considering the wide variety of research projects, scenarios, applications, partners and interaction modalities involved in the work presented in this chapter, the relevance and utility of the proposed framework is made evident. Furthermore, it is important to highlight that these works played a very important role in iteratively setting higher goals and providing contexts for the deployment of novel framework features. The next chapter, the last in this thesis, further discusses this iterative approach and its contribution to the main highlights of this work, a summary of which sets the starting grounds for our vision of future developments.

Chapter 6

Conclusion

Creating multimodal applications presents a tough task, particularly the need to consider each modality to include in the application and deal with each modality's individual challenges. This lead to the motivation of this work, to propose a simple method and provide a multimodal framework to allow application developers to include multiple modalities into their applications without having to worry about the technical aspects of each modality.

The work presented here started with a survey to understand the concepts behind Human Computer Interaction and its evolution, and the basic aspects of multimodal interaction, such as architectures, modalities, fusion and fission of modalities. A search for possible modalities that are more natural to humans was performed, which lead us to conclude that speech is one of the most important, since it is also the main way for humans to communicate. Also, we presented a number of tools and frameworks related to multimodal interaction.

The review of the related work supported the relevance of the motivation and objectives of the work, and a set of initial requirements were determined.

This work was accomplished using an iterative method to develop and create the multimodal interaction framework.

6.1 Main results

The main result of this work is, without a doubt, the multimodal interaction framework and the architecture that enables developers to create multimodal applications more easily by providing the definition of the messages to be exchanged between modalities and defining protocols, so new applications can be made and new modalities created.

The multimodal framework, beyond the definition of protocols, also provides generic modalities, which are ready for integration in different applications. The notable example is the speech modality, an important asset for interaction due to its close relation with how humans naturally communicate, but very complex to implement and deploy given the number of technologies included and its dependence on the target language. The developed speech modality supports multilanguage for both speech recognition and speech synthesis. With this modality a service is also provided capable of automatically translating a base grammar for several languages and providing mechanisms to manually verify and correct it, if needed.

Other contribution is the fusion module, working as part of the interaction manager. This module can combine events from different modalities resulting in a defined action for the application and was created to improve on how fusion is done. Care was taken to propose a method that ensures it does not require much effort to configure: modalities are responsible for announcing their events and, in one line, developers can define the combination of two events.

A great achievement provided by the proposed multimodal interaction framework was the capacity to handle multi-device. Changes introduced to the interaction manager enable multi-device support in a wide variety of scenarios with solutions ranging from a unique interaction manager, in the cloud, to

individual local interaction managers performing discovery in the local network and interconnecting.

6.2 Discussion

In general, the goals of this work (a multimodal architecture and framework that provides easy integration for applications, offering generic modalities, support for multi-device and providing fusion capabilities) have been accomplished, using an iterative method in the process to design and develop the multimodal architecture and framework. Features to meet the requirements have been added at each iteration, supporting each of the objectives set for this thesis.

The work carried out followed an iterative methodology, profiting from different research projects and with a constant evolution of the framework. Figure 6-1 illustrates this iterative nature. Overall, the context of successive projects has provided the onset for specific requirements and goals that led to developments and improvements of the framework to serve concrete multimodal applications. It is important to note that the work carried out is not the product of a single project or supported on a single application. In fact, several projects and applications have profited from our framework proposal and have contributed to its improvement and design. The framework was particularly important for the PaeLife project. This is, we argue an additional proof that our work serves the proposed goals, providing enough flexibility to adapt to the challenging interaction scenarios existing nowadays.

Our expectations for the use of the multimodal framework have been reached: to simplify the work of developers to include multiple modalities into their applications. The applications that have not been developed by the author, the greater part of the personal life assistant AALFred and the Autism application, confirmed those expectations. Insights given by the developers of those applications confirmed that the method to integrate the framework in

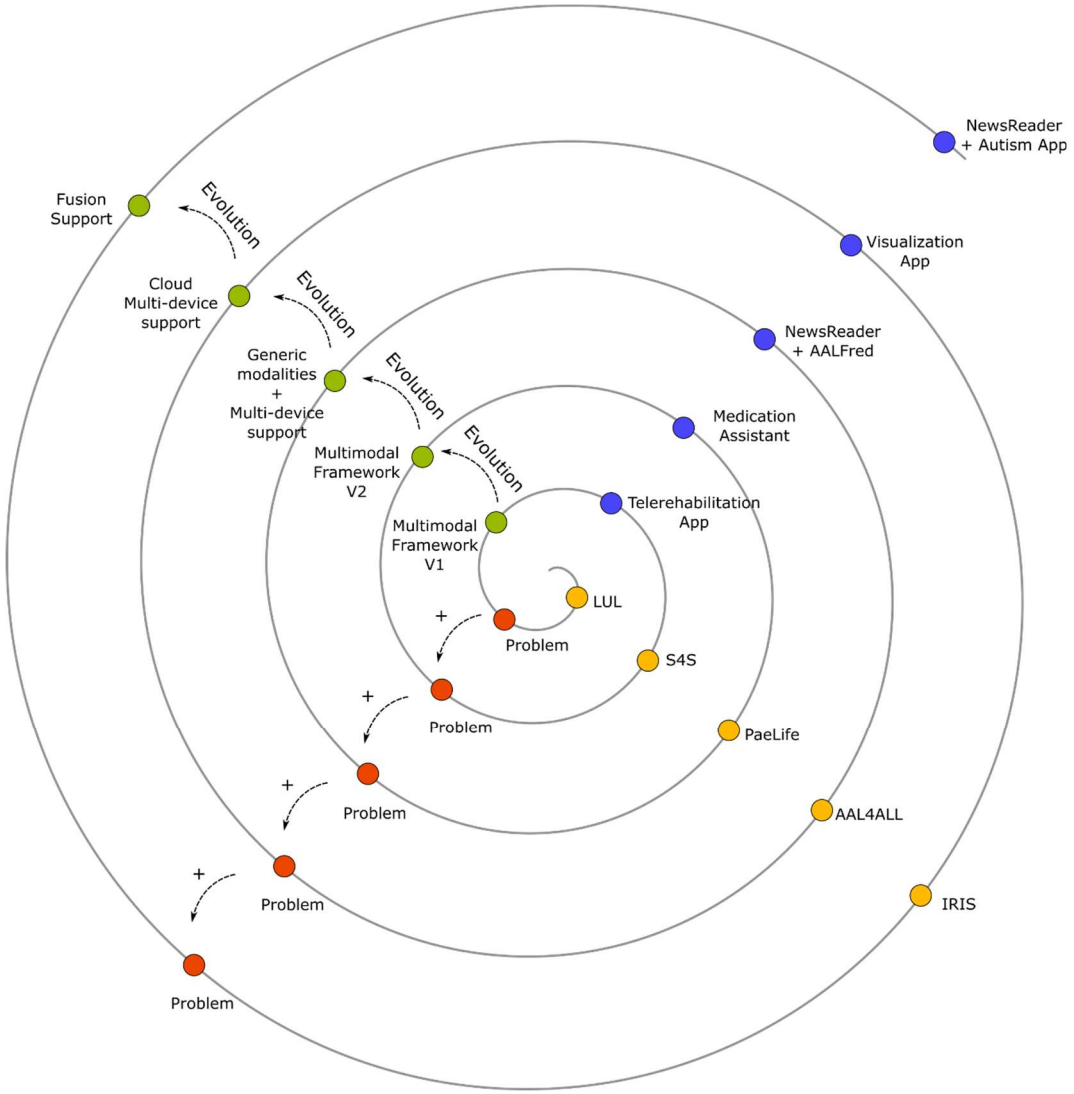

Figure 6-1 – Iterative process to design and develop the multimodal framework. Each stage of the iterative process starts with a project/context then a problem, evolution of the multimodal framework and a demonstration with an application.

applications facilitate their work to adopt multimodal interaction. This work also has contributed for the positive results presented in the involved projects.

An important contribution has also been performed by addressing the issue of modality fusion. A fusion module has been proposed, along with a method for its easy specification and integrating it with the interaction manager. The module currently supports basic fusion operations, and establishes the grounds for more advanced work on the subject.

Considering the results of this work, some important contributions to the state of the art were made, namely, a framework and a method that provide an easy way to integrate multimodality into applications, offering ready to use generic modalities. Additionally, this is provided based on a decoupled architecture, adhering to some of most current standards proposed by the W3C. The extension of the framework to use in multi-device scenarios is also an important contribution, since it allows using one application in several devices seamlessly, both independently or profiting from multiple devices to obtain a richer interaction ecosystem.

6.3 Future Work

The multimodal framework has proven capable to provide multimodality to applications, but further improvements and functionalities can be added. New and more generic modalities can be considered to be developed and added to the multimodal framework.

Further research on fusion engines can be done, to improve the frameworks' fusion engine, by fully supporting, for instance, the CARE properties. Very recent work on the subject can be taken into account such as fusion based on ontologies (Djaid, Saadia, & Ramdane-Cherif, 2015). The current fusion engine combines events and creates new events, but this can evolve to create the unification of the information carried in the combined events, generating new events that are easier to interpret. Also, it is important to assess, with developers, how they use the created method to define the fusion events and its integration in applications.

The developed multimodal framework is presented with a focus on the input side of the interactions. It presents output modalities and how they are managed but lacks some features. A fission module can be created and placed at the output end of the interaction manager. The module could be responsible for selecting the output modalities and can be particularly important in the multi-device scenarios. Additionally, the framework could be endowed with adaptation

features by integrating environment and context variables, in line with recent works showing evolutions on adaptable user interfaces (Varela, Paz-Lopez, Becerra, & Duro, 2016), presenting outputs that are more suited to the users in their environment.

Also the support for multi-device could also be assessed considering other devices such as, for example, smartwatches.

REFERENCES

AAL4ALL-Consortium. (2015). *Relatório Técnico - Científico Março 2011 – Fevereiro 2015*.

Almeida, N., Silva, S., Santos, B. S., & Teixeira, A. (2016). Interactive, Multi-Device Visualization Supported by a Multimodal Interaction Framework: Proof of Concept. In *HCII2016*. Toronto.

Almeida, N., Silva, S., & Teixeira, A. (2014a). Design and Development of Speech Interaction: A Methodology. In *Proceedings of HCI International 2014*.

Almeida, N., Silva, S., & Teixeira, A. (2014b). Multimodal Multi-Device Application Supported by an SCXML State Chart Machine. In *Workshop on Engineering Interactive Systems with SCXML, The sixth ACM SIGCHI Symposium on Computing Systems*.

Almeida, N., Teixeira, A., Rosa, A. F., Braga, D., Freitas, J., Dias, M. S., … Saldanha, N. (2015). Giving Voices to Multimodal Applications. In M. Kurosu (Ed.), *Human-Computer Interaction: Interaction Technologies: 17th International Conference, HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings, Part II* (pp. 273–283). Springer International Publishing.

Atrey, P. K., Hossain, M. A., El Saddik, A., & Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, *16*(6), 345–379. http://doi.org/10.1007/s00530-010-0182-0

Badam, S. K., & Elmqvist, N. (2014). PolyChrome: A Cross-Device Framework for Collaborative Web Visualization. *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces - ITS '14*.

Baggia, P., Burnett, D. C., Carter, J., Dahl, D. A., McCobb, G., & Raggett, D.

(2009). EMMA: Extensible MultiModal Annotation markup language. Retrieved from http://www.w3.org/TR/emma/

Barnett, J., Akolkar, R., Auburn, R. J., Bodell, M., Burnett, D. C., Carter, J., … Roxendal, J. (2015). State Chart XML (SCXML): State Machine Notation for Control Abstraction. Retrieved from http://www.w3.org/TR/scxml/

Bellifemine, F., Caire, G., Poggi, A., & Rimassa, G. (2003). JADE-A White Paper, Sept. 2003. *Online Document Accessible under Http://jade. Tilab. com/papers/2003/WhitePaperJADEEXP. Pdf (Last Access: March 22, 2008)*.

Bernsen, N. O. (2008). Multimodality Theory. In D. Tzovaras (Ed.), *Multimodal User Interfaces - From Signals to Interaction* (pp. 5–29). Springer Berlin Heidelberg.

Bernsen, N. O., & Dybkjær, L. (2010). Multimodal Usability- more on modalities. Retrieved November 20, 2015, from http://www.multimodalusability.dk/mus_modalities.php

Berti, S., & Paternò, F. (2005). Migratory multimodal interfaces in multidevice environments. *Proceedings of the 7th International Conference on Multimodal Interfaces. ACM*.

Blumendorf, M., Roscher, D., & Albayrak, S. (2010). Dynamic user interface distribution for flexible multimodal interaction. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction on - ICMI-MLMI '10* (p. 1). New York, New York, USA: ACM Press. http://doi.org/10.1145/1891903.1891930

Bodell, M., Dahl, D., Kliche, I., Larson, J., & Porter, B. (2012). Multimodal Architecture and Interfaces, W3C. Retrieved from http://www.w3.org/TR/mmi-arch/

Bolt, R. A., & Bolt, R. A. (1980). "Put-that-there." In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques - SIGGRAPH '80* (Vol. 14, pp. 262–270). New York, New York, USA: ACM Press. http://doi.org/10.1145/800250.807503

Bouchet, J., & Nigay, L. (2004). ICARE: a component-based approach for the design and development of multimodal interfaces. In *Extended abstracts of the 2004 conference on Human factors and computing systems - CHI '04* (p. 1325). New York, New York, USA: ACM Press. http://doi.org/10.1145/985921.986055

Bourguet, M.-L. (2003). Designing and Prototyping Multimodal Commands. *INTERACT*.

Bunt, H., Alexandersson, J., Carletta, J., Choe, J.-W., Fang, A. C., Hasida, K., … Traum, D. (2010). Towards an ISO Standard for Dialogue Act Annotation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European

Language Resources Association (ELRA).

Carroll, J. M. (John M. (2002). *Human-computer interaction in the new millennium*. ACM Press.

Cassidy, S. (n.d.). Chapter 14. Language Models in ASR - COMP449: Speech Recognition. Retrieved from http://web.science.mq.edu.au/~cassidy/comp449/html/ch14.html

Chung, H., North, C., Self, J. Z., Chu, S., & Quek, F. (2014). VisPorter: Facilitating Information Sharing for Collaborative Sensemaking on Multiple Displays. *Personal and Ubiquitous Computing*, *18*(5), 1169–1186. http://doi.org/10.1007/s00779-013-0727-2

Cohen, P. R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., … Clow, J. (1997a). QuickSet: multimodal interaction for distributed applications. In *Proceedings of the fifth ACM international conference on Multimedia - MULTIMEDIA '97* (pp. 31–40). New York, New York, USA: ACM Press. http://doi.org/10.1145/266180.266328

Cohen, P. R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., … Clow, J. (1997b). QuickSet: multimodal interaction for simulation set-up and control. In *Proceedings of the Fifth Conference on Applied Natural Language Processing* (pp. 20–24). Morristown, NJ, USA: Association for Computational Linguistics. http://doi.org/10.3115/974557.974562

Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J., & Young, R. M. (1995). Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The Care Properties. In K. Nordby, P. Helmersen, D. J. Gilmore, & S. A. Arnesen (Eds.), *Human—Computer Interaction* (pp. 115–120). Boston, MA: Springer US. http://doi.org/10.1007/978-1-5041-2896-4_19

Cutugno, F., Leano, V. A., Rinaldi, R., & Mignini, G. (2012). Multimodal framework for mobile interaction. In *Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '12* (p. 197). New York, New York, USA: ACM Press. http://doi.org/10.1145/2254556.2254592

Dahl, D. A. (2013). The W3C multimodal architecture and interfaces standard. *Journal on Multimodal User Interfaces*, *7*(3), 171–182. http://doi.org/10.1007/s12193-013-0120-5

Dieter, G. E., & Schmidt, L. C. (2013). *Engineering design*. McGraw-Hill.

Dix, A. (2009). *Human-computer interaction*. Springer US.

Dix, A., Finlay, J., Abowd, G. D., & Beale, R. (1993). *Human-Computer Interaction*. London: Prentice Hall.

Djaid, N. T., Saadia, N., & Ramdane-Cherif, A. (2015). Multimodal Fusion Engine for an Intelligent Assistance Robot Using Ontology. *Procedia Computer Science*, *52*, 129–136. http://doi.org/10.1016/j.procs.2015.05.041

Doyle, J., & Bailey, C. (2014). Lessons learned in deploying independent living

technologies to older adults' homes. *Universal Access in the Information Society*, *13*(2), 191–204.

Dragicevic, P., & Fekete, J.-D. (2001). Input device selection and interaction configuration with ICON. *People and Computers XV—Interaction without Frontiers*, 543–558.

Dumas, B., Ingold, R., & Lalanne, D. (2009). Benchmarking fusion engines of multimodal interactive systems. In *Proceedings of the 2009 international conference on Multimodal interfaces - ICMI-MLMI '09* (pp. 169–176). New York, New York, USA: ACM Press. http://doi.org/10.1145/1647314.1647345

Dumas, B., Lalanne, D., Guinard, D., Koenig, R., & Ingold, R. (2008). Strengths and weaknesses of software architectures for the rapid creation of tangible and multimodal interfaces. *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction - TEI '08*, 47–54.

Dumas, B., Lalanne, D., & Ingold, R. (2008). Prototyping multimodal interfaces with smuiml modeling language. *CHI 2008 Workshop on User Interface Description Languages for Next Generation User Interfaces, CHI*, 63–66.

Dumas, B., Lalanne, D., & Ingold, R. (2009). HephaisTK: a toolkit for rapid prototyping of multimodal interfaces. In *Proceedings of the 2009 international conference on Multimodal interfaces - ICMI-MLMI '09* (pp. 231–232). New York, New York, USA: ACM Press. http://doi.org/10.1145/1647314.1647360

Dumas, B., Lalanne, D., & Oviatt, S. (2009). Multimodal Interfaces: A Survey of Principles, Models and Frameworks. In L. Denis & K. Jürg (Eds.), *Human Machine Interaction* (pp. 3–26). Springer-Verlag.

Eisma, R. (2003). Mutual inspiration in the development of new technology for older people. In *Proceeding INCLUDE 2003*. London, UK.

Ferreira, F., Almeida, N., Rosa, A. F., Oliveira, A., Casimiro, J., Silva, S., & Teixeira, A. (2014). Elderly Centered Design for Interaction – The Case of the S4S Medication Assistant. *Procedia Computer Science*, *27*, 398–408. http://doi.org/10.1016/j.procs.2014.02.044

Ferreira, F., Almeida, N., Rosa, A. F., Oliveira, A., Teixeira, A., & Pereira, J. C. (2013). Multimodal and adaptable medication assistant for the elderly: A prototype for interaction and usability in smartphones. In *Information Systems and Technologies (CISTI), 2013 8th Iberian* (pp. 1–6). Lisboa: IEEE.

Ghangurde, M. (2010). Ford SYNC and Microsoft Windows Embedded Automotive Make Digital Lifestyle a Reality on the Road. *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, *3*(2010-01–2319), 99–105.

Gómez, D. (2015). Little arrangements that matter. Rethinking autonomy-enabling innovations for later life. *Technological Forecasting and Social*

Change, *93*, 91–101.

Hak, R., Dolezal, J., & Zeman, T. (2012). Manitou: A multimodal interaction platform. In *2012 5th Joint IFIP Wireless and Mobile Networking Conference (WMNC)* (pp. 60–63). IEEE. http://doi.org/10.1109/WMNC.2012.6416147

Hämäläinen, A., Teixeira, A., Almeida, N., Meinedo, H., Fegyó, T., & Dias, M. S. (2015). Multilingual speech recognition for the elderly: The AALFred personal life assistant. *Procedia Computer Science*, *67*, 283–292.

Hamilton, P., & Wigdor, D. J. (2014). Conductor: enabling and understanding cross-device interaction. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (pp. 2773–2782). New York, New York, USA: ACM Press. http://doi.org/10.1145/2556288.2557170

Heikkinen, T., Goncalves, J., Kostakos, V., Elhart, I., & Ojala, T. (2014). Tandem Browsing Toolkit : Distributed Multi - Display Interfaces with Web Technologies. *Proceedings of The International Symposium on Pervasive Displays*, 142–147.

Hewett, T. T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., … Verplank, W. (1992). *ACM SIGCHI curricula for human-computer interaction*. Association for Computing Machinery.

Hinckley, K., & Wigdor, D. (2002). Input technologies and techniques. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, 151–168.

Holzinger, A. (2003). Finger Instead of Mouse: Touch Screens as a Means of Enhancing Universal Access. In N. Carbonell & C. Stephanidis (Eds.), *Universal Access. Theoretical Perspectives, Practice, and Experience* (pp. 387–397). Berlin, Heidelberg: Springer Berlin Heidelberg.

Hoste, L., Dumas, B., & Signer, B. (2011). Mudra: a unified multimodal interaction framework. In *Proceedings of the 13th international conference on multimodal interfaces - ICMI '11* (p. 97). New York, New York, USA: ACM Press. http://doi.org/10.1145/2070481.2070500

Houben, S., & Marquardt, N. (2015). WATCHCONNECT: A Toolkit for Prototyping Smartwatch-Centric Cross-Device Applications. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 1247–1256.

Hunt, A., & McGlashan, S. (2014). Speech Recognition Grammar Specification Version 1.0. Retrieved March 14, 2015, from http://www.w3.org/TR/speech-grammar/

Jaimes, A., & Sebe, N. (2007). Multimodal human - computer interaction: A survey. *Computer Vision and Image Understanding*, *108*(1–2), 116–134.

Jana, A. (2012). *Kinect for Windows SDK Programming Guide*. Packt

Publishing.

Johanson, B., Fox, A., & Winograd, T. (2002). The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. *IEEE Pervasive Computing*, *1*(2), 67–74.

Johnston, M., Cohen, P. R., McGee, D., Oviatt, S. L., Pittman, J. A., Smith, I., … Smith, I. (1997). Unification-based multimodal integration. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics* - (pp. 281–288). Morristown, NJ, USA: Association for Computational Linguistics. http://doi.org/10.3115/979617.979653

Johnston, M., Fabbrizio, G. Di, & Urbanek, S. (2011). mTalk - A Multimodal Browser for Mobile Services. In *12th Annual Conference of the International Speech Communication Association, INTERSPEECH 2011* (pp. 3261–3264). ISCA.

Jurafsky, D., & Martin, J. H. (2000). HMMs and Speech Recognition. In *Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall.

Karray, F., Alemzadeh, M., Saleh, J. A., & Arab, M. N. (2008). Human-computer interaction: Overview on state of the art. *International Journal On Smart Sensing and Intelligent Systems*, *1*(1), 137–159.

Kernchen, R., Meissner, S., Moessner, K., Cesar, P., Vaishnavi, I., Boussard, M., & Hesselman, C. (2010). Intelligent Multimedia Presentation in Ubiquitous Multidevice Scenarios. *IEEE Multimedia*, *17*(2), 52–63. http://doi.org/10.1109/MMUL.2009.75

Lawson, J.-Y. L., Al-Akkad, A.-A., Vanderdonckt, J., & Macq, B. (2009). An open source workbench for prototyping multimodal interactions based on off-the-shelf heterogeneous components. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems - EICS '09* (pp. 245–254). New York, New York, USA: ACM Press. http://doi.org/10.1145/1570433.1570480

Lewis, J. R., & Sauro, J. (2009). The Factor Structure of the System Usability Scale. In M. Kurosu (Ed.), *Human Centered Design Lecture Notes in Computer Science* (Vol. 5619, p. pp 94--103). Berlin, Heidelberg: Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-642-02806-9

Liu, Y., Cornish, A., & Clegg, J. (2007). ICT and Special Educational Needs: Using Meta-synthesis for Bridging the Multifaceted Divide. In Y. Shi, G. D. van Albada, J. Dongarra, & P. M. A. Sloot (Eds.), *Computational Science – ICCS 2007* (Vol. 4490, pp. 18–25). Berlin, Heidelberg: Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-540-72590-9

Martins, A., Queirós, A., & Cerqueira, M. (2012). The International

Classification of Functioning, Disability and Health as a conceptual model for the evaluation of environmental factors. *Proceedings of the 4th International Conference on Software Development for Enhancing Accessibility and Fighting Info-Exclusion (DSAI 2012)*, *14*, 293–300.

Martins, C. (2008). *Dynamic Language Modeling for European Portuguese*. University of Aveiro.

Matlabi, H., Parker, S., & McKee, K. (2011). The contribution of home-based technology to older people's quality of life in extra care housing. *BMC Geriatrics*.

Mcglaun, G., Mcglaun, G., Lang, M., & Rigoll, G. (2004). Development of a Generic Multimodal Framework for Handling ErrorPatternsduring Human-Machine Interaction. *SCI 2004, 8th World Multi-Conference on Systems, Cybernetics*.

Michael Bodell, Dahl, D., Larson, J., Porter, B., Raggett, D., Raman, T. V., … Moshe Yudkowsky. (2012). Multimodal Architecture and Interfaces, W3C. Retrieved November 20, 2013, from http://www.w3.org/TR/mmi-arch/

Microsoft Corporation. (2013). *Kinect for Windows - Human Interface Guidelines*.

Möller, A., Diewald, S., Roalter, L., & Kranz, M. (2014). M3I: A Framework for Mobile Multimodal Interaction. *Mensch & Computer*.

Moschetti, A., Fiorini, L., Aquilano, M., Cavallo, F., & Dario, P. (2014). Preliminary findings of the AALIANCE2 ambient assisted living roadmap. In *Ambient Assisted Living* (pp. 335–342). Springer.

Mynatt, E., & Rogers, W. (2001). Developing technology to support the functional independence of older adults. *Ageing International*, *27*(1), 24–41.

Newell, A., Arnott, J., Carmichael, A., & Morgan, M. (2007). Methodologies for Involving Older Adults in the Design Process. In C. Stephanidis (Ed.), *Universal Acess in Human Computer Interaction. Coping with Diversity* (Vol. 4554). Berlin, Heidelberg: Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-540-73279-2

Nigay, L., & Coutaz, J. (1993). A design space for multimodal systems. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '93* (pp. 172–178). New York, New York, USA: ACM Press. http://doi.org/10.1145/169059.169143

Niklfeld, G., Finan, R., & Pucher, M. (2001). Architecture for adaptive multimodal dialog systems based on voiceXML. In *INTERSPEECH* (pp. 2341–2344).

Norris, S. (2004). *Analyzing multimodal interaction: A methodological framework*. (Routledge).

Organization, W. H. (2002). Active ageing: a policy framework. In *Second United*

Nations World Assembly on Ageing. Madrid, Spain.

Oviatt, S. (2003). Multimodal interfaces. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, Second Edition*, 413–432.

Paternò, F. (2004). Multimodality and Multi-device interfaces. In *W3C Workshop on Multimodal Interaction*. Sophia Antipolis.

Pereira, C., Almeida, N., Martins, A. I., Silva, S., Rosa, A. F., Silva, M. O. e, & Teixeira, A. (2015). Evaluation of Complex Distributed Multimodal Applications Evaluating a TeleRehabilitation System When It Really Matters. In *Human Aspects of IT for the Aged Population. Design for Everyday Life*.

Pereira, J. C., & Teixeira, A. (2015). Geração de Linguagem Natural para Conversão de Dados em Texto: Aplicação a um Assistente de Medicação para o Português [Trainable NLG for Data to Portuguese – With application to a Medication Assistant]. *LinguaMática*, *7*(1).

Pereira, J. C., Teixeira, A., & Pinto, J. (2012). Natural Language Generation in the context of Multimodal Interaction in Portuguese. *Electrónica E Telecomunicações*, *5*(4).

Pinelle, D., Gutwin, C., & Greenberg, S. (2003). Task analysis for groupware usability evaluation. *ACM Transactions on Computer-Human Interaction*, *10*(4), 281–311. http://doi.org/10.1145/966930.966932

Pocheville, A., Kheddar, A., & Yokoi, K. (2004). I-touch: a generic multimodal framework for industry virtual prototyping. In *IEEE Conference on Robotics and Automation, 2004. TExCRA Technical Exhibition Based.* (pp. 65–66). IEEE. http://doi.org/10.1109/TEXCRA.2004.1425000

Pous, M., & Ceccaroni, L. (2010). Multimodal Interaction in Distributed and Ubiquitous Computing. In *2010 Fifth International Conference on Internet and Web Applications and Services* (pp. 457–462). IEEE. http://doi.org/10.1109/ICIW.2010.75

Preece, J., Paine, L., & Rogers, Y. (2015). *Interaction Design-beyond human-computer interaction* (Fourth Ed.). Wiley.

Reeves, L. M., Lai, J., Larson, J. A., Oviatt, S., Balaji, T. S., Buisine, S., … Wang, Q. Y. (2004). Guidelines for multimodal user interface design. *Communications of the ACM - Multimodal Interfaces That Flex, Adapt, and Persist*, *47*(1), 57–59.

Reiter, E., & Dale, R. (2006). *Building Natural Language Generation Systems*. (E. Reiter & R. Dale, Eds.). Cambridge University Press.

Revuelta-Martínez, A., Rodríguez, L., García-Varea, I., & Montero, F. (2013). Multimodal interaction for information retrieval using natural language. *Computer Standards & Interfaces*, *35*(5), 428–441.

Saffer, D. (2008). *Designing gestural interfaces*. OReilly Media.

Schnelle-Walka, D., Radomski, S., & Mühlhäuser, M. (2014). Multimodal Fusion and Fission within W3C Standards for Nonverbal Communication with Blind Persons. In *Computers Helping People with Special Needs* (pp. 209–213). Springer International Publishing. http://doi.org/10.1007/978-3-319-08596-8_33

Serrano, M., Nigay, L., Lawson, J.-Y. L., Ramsay, A., Murray-Smith, R., & Denef, S. (2008). The openinterface framework: a tool for multimodal interaction. In *Proceeding of the twenty-sixth annual CHI conference extended abstracts on Human factors in computing systems - CHI '08* (p. 3501). New York, New York, USA: ACM Press. http://doi.org/10.1145/1358628.1358881

Seyed, A. (2013). Examining User Experience in Multi-Display Environments. Doctoral dissertation, University of Calgary. Computer Science.

Shen, C., Esenther, A., Forlines, C., & Ryall, K. (2006). Three modes of multisurface interaction and visualization. In *Information Visualization and Interaction Techniques for Collaboration across Multiple Displays Workshop associated with CHI* (Vol. 6).

Sinha, A. K., & Landay, J. A. (2003). Capturing user tests in a multimodal, multidevice informal prototyping tool. *Proceedings of the 5th International Conference on Multimodal Interfaces, ICMI '03*, 117–124.

Sonntag, D., Engel, R., Herzog, G., Pfalzgraf, A., Pfleger, N., Romanelli, M., & Reithinger, N. (2007). SmartWeb Handheld — Multimodal Interaction with Ontological Knowledge Bases and Semantic Web Services. In *Artifical Intelligence for Human Computing* (pp. 272–295). Berlin, Heidelberg: Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-540-72348-6_14

Steg, H., Strese, H., Loroff, C., Hull, J., & Sophie Schmidt. (2006). Ambient Assisted Living–European Overview Report, March 2006. *EU Specific Support Action*.

Tadeusiewicz, R. (2010). Speech in human system interaction. In *3rd International Conference on Human System Interaction* (pp. 2–13). IEEE. http://doi.org/10.1109/HSI.2010.5514597

Tashev, I. (2013). Kinect Development Kit: A Toolkit for Gesture- and Speech-Based Human-Machine Interaction [Best of the Web]. *IEEE Signal Processing Magazine*, *30*(5), 129–131. http://doi.org/10.1109/MSP.2013.2266959

Taylor, P. (2009). *Text-to-Speech Synthesis* (1st Editio). Cambridge University Press.

Teixeira, A. (2014). A Critical Analysis of Speech-based Interaction in Healthcare Robots: Making a Case for the Increased Use of Speech in Medical and Assistive Robots. In A. Neustein (Ed.), *Speech and Automata*

in Healthcare : Voice-Controlled Medical and Surgical Robots (pp. 1–29). De Gruyter.

Teixeira, A., Braga, D., Coelho, L., & Fonseca, A. (2009). Speech as the basic interface for assistive technology. In *International Conference on Software Development for Enhancing Accessibility and Fighting Info-Exclusion*. Porto Salvo.

Teixeira, A., Ferreira, F., Almeida, N., Rosa, A. F., Casimiro, J., Silva, S., … Oliveira, A. (2013). Multimodality and Adaptation for an Enhanced Mobile Medication Assistant for the Elderly. In *Third Mobile Accessibility Workshop (MOBACC), CHI*. France, April.

Teixeira, A., Ferreira, F., Almeida, N., Silva, S., Rosa, A. F., Pereira, J. C., & Vieira, D. (2016). Design and Development of Medication Assistant Elderly-centred Design to Go Beyond Simple Medication Reminders. *Universal Access in the Information Society*.

Teixeira, A., Francisco, P., Almeida, N., Pereira, C., & Silva, S. (2014). Services to Support Use and Development of Speech Input for Multilingual Multimodal Applications for Mobile Scenarios. In *The Ninth International Conference on Internet and Web Applications and Services (ICIW 2014),Track WSSA - Web Services-based Systems and Applications*.

Teixeira, A., Francisco, P., Almeida, N., Pereira, C., & Silva, S. (2015). Services to Support Use and Development of Multilingual Speech Input. *International Journal On Advances in Internet Technology*, *8*(1&2), 1–12.

Teixeira, A., Hämäläinen, A., Avelar, J., Almeida, N., Németh, G., Fegyó, T., … Dias, M. S. (2013). Speech-Centric Multimodal Interaction for Easy-To-Access Online Services : A Personal Life Assistant for the Elderly. In *Proc. DSAI 2013, Procedia Computer Science* (pp. 389–397).

Teixeira, A., Pereira, C., Silva, M. O. e, Pacheco, O., Neves, A., & Casimiro, J. (2011). AdaptO - Adaptive Multimodal Output. In *Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems* (pp. 91–100). Vilamoura.

Teixeira, A., Pereira, J. C., Francisco, P. G., & Almeida, N. (2015). Tradução Automática na Interação com Máquinas. In *Oslo Studies in Language* (Vol. 7).

Tobii: this is eye tracking. (2015). Retrieved November 10, 2016, from http://www.tobii.com/group/about/this-is-eye-tracking/

Tripathi, K. (2011). A study of interactivity in human computer interaction. *International Journal of Computer Applications*, *16*(6).

United Nations, Department of Economic and Social Affairs, P. D. (2013). (n.d.). *World Population Ageing 2013*. *World Population Ageing 2013*. ST/ESA/SER.A/348.

Varela, G., Paz-Lopez, A., Becerra, J. A., & Duro, R. (2016). A Framework for the Development of Context-Adaptable User Interfaces for Ubiquitous Computing Systems. *Sensors (Basel, Switzerland)*, *16*(7). http://doi.org/10.3390/s16071049

Vieira, D. (2015). *Enhanced Multimodal Interaction Framework and Applications. Master thesis, Aveiro, Universidade de Aveiro*. Universidade de Aveiro.

Vieira, D., Freitas, J. D., Acartürk, C., Teixeira, A., Sousa, L., Silva, S., … Dias, M. S. (2015). Read That Article: Exploring Synergies between Gaze and Speech Interaction, 341–342. http://doi.org/10.1145/2700648.2811369

Ward, W. (1991). Understanding spontaneous speech: the Phoenix system. In *International Conference on Acoustics, Speech, and Signal Processing, 1991. ICASSP-91.* (Vol. 1, pp. 365–367). IEEE Computer Society.

Weibel, N., Satyanarayan, A., Lazar, A., Oda, R., Yamaoka, S., Doerr, K.-U., … Hollan, J. D. (2011). Hiperface: a multichannel architecture to explore multimodal interactions with ultra-scale wall displays. In *ICSE'11: Proceedings of the 33rd International Conference on Software Engineering*.

Williams, P., Jamali, H. R., & Nicholas, D. (2006). Using ICT with people with special education needs: what the literature tells us. *Aslib Journal of Information Management*, *58*(4), 330–345. http://doi.org/10.1108/00012530610687704

Woźniak, P., Lischke, L., Schmidt, B., Zhao, S., & Fjeld, M. (2014). Thaddeus : A Dual Device Interaction Space for Exploring Information Visualisation. *Proceedings of the 8th Nordic Conference on Human-Computer Interaction Fun, Fast, Foundational - NordiCHI '14*, 41–50. http://doi.org/10.1145/2639189.2639237

Zygmunt, A., Olfson, M., Boyer, C., & Mechanic, D. (2002). Interventions to improve medication adherence in schizophrenia. *Am. J. Psychiatry*, *159*(10), 1653–1664.