UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

ADAPTATION OF MULTIMODAL INPUT

Pedro Filipe Matos Feiteira

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2011

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

ADAPTATION OF MULTIMODAL INPUT

Pedro Filipe Matos Feiteira

DISSERTAÇÃO

Projecto orientado pelo Prof. Doutor Carlos Alberto Pacheco dos Anjos Duarte

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2011

Acknowledgments

In first place, I would like to thank my advisor, Professor Carlos Alberto Pacheco dos Anjos Duarte, not only for the opportunity of working in such a relevant project but also for his continuous support during my work this past year, which included making countless document reviews, exchanging ideas with me and creating a good work environment in our group.

Other people that must be acknowledged are the remaining members of the ”GUIDE working group”that worked very much throughout this year to make our goals come true: David Costa, Daniel Costa and José Coelho.

To all my colleagues in the Absolute Interaction (AbsInt) group that also contributed to create an enjoyable work environment.

To my family and friends

Resumo

Esta tese tem um forte foco em sistemas multimodais e respectivos módulos de fusão. O trabalho realizado ao longo deste ano está em quase toda a sua maioria relacionado com o projecto europeu científico GUIDE (Gently User Interfaces for Elderly and Disabled Citizens). Os resultados obtidos deste trabalho contribuiram significativamente para o desenvolvimento do projecto e alguma parte continuará a ser desenvolvida no decorrer do próximo ano.

O desenvolvimento de aplicações multimodais pode ser por vezes um processo complexo devido ao número de dispositivos de entrada e saída existentes e o tipo de modalidades disponíveis para interagir. Tornar aplicações accessíveis é normalmente uma tarefa que exige esforço, tempo, e recursos aos desenvolvedores, tornando-a bastante negligenciada. Um segmento da população que é fortemente afectado por este facto são utilizadores idosos, os quais, na sua maioria, sofrem de algum tipo de limitação fisica ou cognitiva.

O objectivo do projecto GUIDE é desenvolver uma *toolbox* de interfaces multimodais adaptativas direccionada para os problemas de accessibilidade apresentados por utilizadores idosos. Esta *framework* irá diminuir o esforço necessário por parte dos desenvolvedores de aplicações em implementar técnicas de accessibilidade. As aplicações que irão ser executadas na *framework* GUIDE são automaticamente adaptadas às necessidades e limitações de cada utilizador. Nesta tese, são apresentadas três aplicações que foram desenvolvidas ao longo deste ano no âmbito do projecto GUIDE.

A UTA (User Trials Application) é uma aplicação multimodal que foi desenhada, implementada e usada para efectuar o levantamento de requisitos e preferências de utilizador, um processo ao qual foi dada bastante enfâse nos primeiros meses do projecto. As tarefas realizadas pelos utilizadores ao longo das várias sessões de testes, involviam diferentes modalidades tais como visão, audição ou cognição. A UTA, como sistema multimodal que é, permite o uso de diferentes meios de entrada e saída de maneira a testar todas as modalidades pretendidas. Um dos aspectos fundamentais desta aplicaçação é o seu elevado grau de customização, o qual permite fácil e flexivelmente definir os testes a serem realizados, o que inclui controlar variáveis tais como o tipo de elementos interactivos que devem surgir no ecrã e as suas propriedades. Outra importante característica da UTA, é incluir uma aproximação baseada na técnica *Wizard-of-Oz*, proporcionando um certo nível

de controlo ao indivíduo que supervisiona a sessão de testes, dando-lhe a hipótese de gerir a execução da aplicação ou o registo de resultados. Ambas as tarefas mencionadas são automaticamente realizadas pela aplicação, mas para uma maior eficácia no levantamento de requisitos e preferências são também auxiliadas pelo *wizard*.

A segunda aplicação desenvolvida nesta tese foi a UIA (User Initialization Application). Esta aplicação funcionou como um protótipo da versão final que irá estar presente dentro da *framework* GUIDE cujo objectivo é servir como um primeiro contacto do utilizador com o sistema. Este objectivo tem dois fins. O primeiro é através de uma série de ecrãs informativos dar ao utilizador uma noção de como fazer uso dos dispositivos de entrada à sua disposição. O segundo fim desta aplicação é, através de uma série de tarefas a realizar, capturar informação sobre o utilizador, em termos das suas capacidades e limitações, e automaticamente atribuir-lhe um modelo de utilizador que irá servir como referência para adaptação.

A UIA inclui diversos testes que abrangem várias modalidades de entrada e saída. Este protótipo, para além de mostrar exemplos de testes que podem ser realizados para caracterizar um utilizador, demonstra também a importância da adaptação em aplicações multimodais. Ao longo da execução do protótipo, à medida que o utilizador interage com a aplicação demonstrando as suas preferências, esta é capaz de se auto-adaptar dinamicamente alterando variáveis tais como tamanho de letra, distância entre botões ou volume.

A última fase desta tese concentra-se em descrever o desenvolvimento do módulo de fusão a ser integrado dentro da *framework* GUIDE. Este componente tem a responsabilidade de combinar entradas multimodais geradas por utilizadores e gerar uma interpretação a partir desses eventos. A análise de resultados observados durante o período de testes em que a UTA foi utilizada, permitiu concluir que os utilizadores quando interagem de forma multimodal, diferem entre si, na medida em que pode existir utilizadores que prefiram combinar modalidades de uma certa maneira ou de outra. Este facto trouxe um reforço à necessidade da existência de fusão num sistema multimodal como é o caso do GUIDE.

A aproximação arquitectural escolhida para implementar fusão de entradas no GFC (Guide Fusion Core) é baseada em *frames*, estruturas de dados que neste contexto, uma vez activados, despoletam o envio de acções ou respostas para outros componentes da *framework*, o que pode provocar uma mudança de estado de uma aplicação. Um *frame* contém um conjunto de condições correspondentes a determinadas modalidades e um conjunto de respostas. Cada *frame* pode ser visto como uma sequência de acções que no contexto actual da aplicação deverá gerar uma determinada resposta pelo sistema. Neste documento é dado um certo foco aos componentes que interagem directamente com o módulo de fusão, de maneira a perceber a sua relação e os tipos de eventos que são trocados entre eles. O processo de criação de frames necessita de ter uma noção dos elementos interactivos que estão a qualquer momento, disponíveis ao utilizador. Este requisito é suportado pela capacidade de o módulo de fusão receber e analisar uma representação

concreta da interface referente ao estado actual da aplicação. Este processo é algo que é expectável que ocorra múltiplas vezes durante o ciclo de vida de uma aplicação, à medida que o estado desta se altera.

Outros dos principais aspectos sobre o módulo de fusão discutido nesta tese é a sua capacidade de adaptação. Muitos dos componentes da *framework* GUIDE possuem comportamentos adaptativos que são geridos por si próprios mas também auxiliados por outros componentes. Por um lado os principais factores que governam a adaptação feita dentro do módulo de fusão são os eventos de entrada fornecidos pelos diferentes reconhecedores do sistema e informações retiradas do modelo de utilizador que retratam a aptitude do utilizador no uso de diversas modalidades. Por outro lado, o módulo de fusão também é susceptível de desencadear adaptação em outros componentes tais como reconhecedores (e.g. enviando os comando disponíveis para determinado contexto da aplicação) ou componentes centrais da *framework* (Dialogue Manager) que ao receber interpretações das acções dos utilizadores alteram o estado da aplicação. A aproximação escolhida para implementar adaptação no GFC foi uma aproximação baseada em pesos, que permite à arquitectura baseada em *frames* usar o modelo de utilizador para garantir que a activação de *frames* não depende só da fiabilidade dos eventos de entrada recebidos mas também das características do utilizador que são traduzidas para valores de confiança (pesos).

Uma das principais lacunas no desenvolvimento de sistemas multimodais é a sua falta de avaliação. Apesar de a implementação actual do módulo de fusão e respectivas estratégias adaptativas estarem ainda no ínicio do seu desenvolvimento, já se começou a dar atenção a métodos de avaliação que possam medir a performance do GFC em termos de eficácia e tempo de resposta. A solução que está a ser desenvolvida a par do GFC, é uma *framework* de avaliação que permite simular o envio de eventos de entradas e controlar os seus parâmetros mais relevantes tais como por exemplo instantes de inicío e fim, conteúdo semântico ou instante de chegada.

Palavras-chave: GUIDE, Fusão Multimodal, UTA, UIA, Adaptação

Abstract

This thesis is strongly coupled with the European project GUIDE (Gentle User Interfaces for Elderly Citizens) which intends to deliver a toolbox of adaptive multimodal interfaces to run on TV set-top boxes. The goal of this framework is to address some of the limitations and disabilities shown by elderly users and automatically adapt web-based applications to their needs also freeing the developers of the need of tackling accessibility issues.

The User Trials Application is a multimodal application that was designed to perform user trials, which consisted on observing the users interacting with a multimodal system that supported multiple input/output modalities and capturing data about this interaction. This application allowed an high customization regarding tests including which interactive elements should appear on screen and their properties. A Wizard-of-Oz technique was used to empower the person running the tests and to allow a greater degree of control and information gathering.

A second application developed, the User Initialization Application, constituted a prototype of the final version that is going to be present in the GUIDE framework, aimed for introducing the user to the system and input devices as well as gathering information about the user limitations so it could be assigned to a specific user model. The tests included in the prototype used various modalities such as speech and gestures. One of the main features of this application is the use of adaptation throughout the test sequence, changing properties such as volume, text size, color, among others.

The third application discussed in this thesis is the GUIDE Fusion Core, responsible for user-adapted input combination. A frame-based algorithm was used to combine information and a weight-based approach to imprint adaptive behaviour into it. Although the implementation of the GUIDE Fusion core is still in its early development, some focus was given to designing an evaluation framework capable of measuring, according to some metrics, the performance of the fusion core.

Keywords: GUIDE, Multimodal Fusion, UTA, UIA, Adaptation

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This introduction will serve as an overview of the work developed in this dissertation scope. First of all, there will be a discussion about the motivation behind all the work that has been done and related research projects. Afterwards, a more detailed overview of the GUIDE project is given, because of its strong relation with all the work present in this thesis. This section will conclude with a list of contributions made, the results achieved and an explanation about how this document is structured.

1.1 Motivation and Related Research Projects

Nowadays, aging and accessibility are two subjects that are highly correlated in many contexts, including interaction with computers. It is a known fact that around half of the elder population suffers of some kind of disability such as motor impairment, which poses problems and challenges to social interaction [12, 10]. For such end-users, accessible interfaces can make much more of a difference in living quality than for any other citizens.

Multimodal interfaces, by allowing the use of multiple modalities, offer their users the chance of having a natural, more "human" way of interacting. By using modalities like voice or gestures, the communication between user and machine, becomes closer to what people are used to in human-human interaction. This aspect is even more relevant when the user group is composed of elderly people, whom can possess one or several types of impairments. Therefore it would be beneficial to this specific target group, the possibility of using the modalities they are most used to, or the ones with which they are most effective with. Users with hearing impairments could, for example, use visual modalities to interact. This flexibility shown by multimodal interfaces is of great importance to users [4] because it could potentially boost their inclusion in their surrounding private and professional communities; however, it's not expected from them to execute certain selection and configuration operations on this kind of systems. For this reason, the adoption of adaptive multimodal interfaces becomes a solution to consider. If these adaptive features are correctly implemented in user interfaces, it would allow disabled and elderly people

to interact with applications in a more intuitive and supportive manner.

Application developers can also benefit greatly from this adaptation, because as it is today, implementation of accessible user interfaces is still expensive and risky, thanks to the effort developers must spend, thinking about how to cope with user-specific needs and limitations, as well as possessing the experience and knowledge to deal with technological challenges brought by innovative user interfaces approaches. These difficulties in designing applications with accessibility on mind, makes their implementation simply neglect special needs and exclude a large portion of their potential users. If a system is able to perform self-adaptation to enhance the overall accessibility and automatically fit to the individual requirements of users with different kinds of impairments then some part of the developers work and issues can be eased.

The work developed on this thesis is strongly coupled with a European research project named GUIDE (Gently User Interfaces for Disabled and Elderly Citizens) [9] which was conceived in order to tackle the problem just stated above. The next section will elaborate a little more on this project, including its characteristics, and how the work done in this dissertation relates to it.

1.2 GUIDE

In this section several aspects of the GUIDE project will be considered, such as an overview of the system, its main goals, the type of users involved, the input devices and mechanisms to be implemented as well as their respective modalities, the different forms of output responses and feedback to the users. The sections concludes by stating the role assigned to both FCUL and myself in the project development.

1.2.1 Overview

The GUIDE framework can be seen as being composed by three major components; input, output and adaptation, which is a central feature of the framework, as we can see in Figure 1.1.

The input interpretation is the fusion of the input from the different input components and its later translation to a command understandable by the developer application. A dialog manager (not represented in the figure) is responsible for defining the language and handle communication between components (including applications). The output generation has the task of selecting the proper output modalities, and distribute content among them. The adaptation module by storing and constant updating information about users, is able to assist: individual input modules (e.g. changing the settings of a video recognizer by setting different camera views); the posterior interpretation regarding input fusion (e.g. setting different weights for modalities according to a specific user).

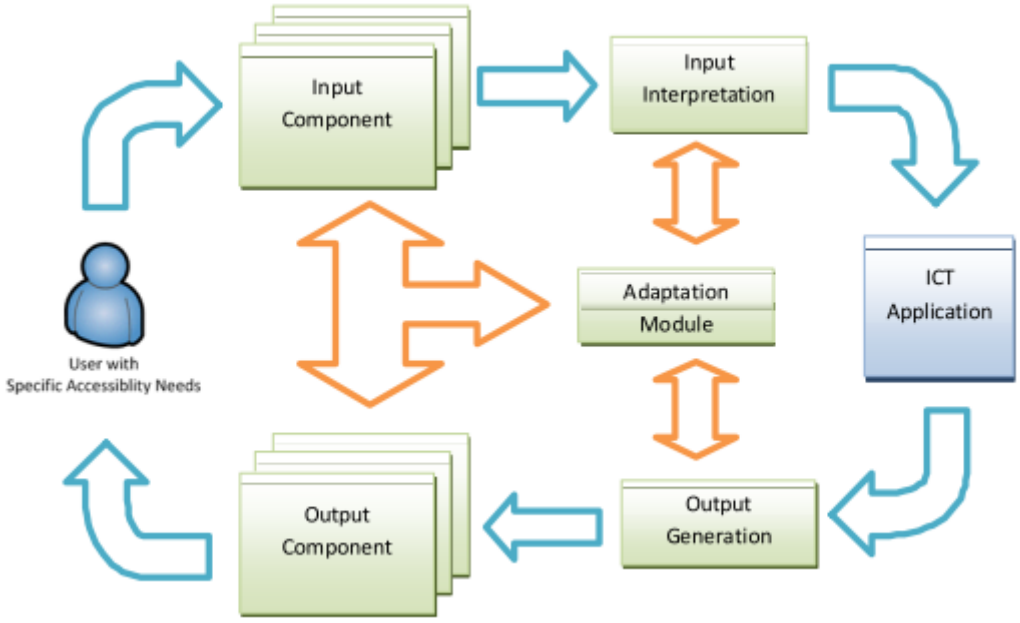


Figure 1.1: High level view of GUIDE architecture and flow of information (Blue arrows represent data and orange arrows represent control information

1.2.2 Aims

The main intent of the European project GUIDE is to develop a toolbox of adaptive, multimodal user interfaces that target the accessibility requirements of elderly users in their home environment, making use of TV STBs (set-top boxes) as processing and connectivity platform. With its included software, hardware and documentation, this toolbox will put developers of ICT applications in the position to easier implement truly accessible applications using the most recent user interface technologies with reduced development risk, cost and time to market. For this purpose, the toolbox will provide not only the technology of advanced multi-modal UI components but also the adaptation mechanisms necessary to let the UI components interoperate with legacy and novel applications, including the capability to self-adapt to end-user needs. Along with this toolbox, the project GUIDE also aims to deliver other items of relevant importance which include a standardized user model that reflects impairments and preferences of elderly users, enabling an user-driven adaptation; a “virtual user”-centred design process oriented for developers, so that the involvement of user requirements in the development phase can continually grow; relevant design knowledge for application developers; knowledge about interaction patterns of the elderly related to ageing, impairments and preferences through a series of extensive tests; reference applications (e.g. home automation, video conferencing, tele-learning) that improve social inclusion, assisted living and continuous learning of elderly users.

1.2.3 Users

As stated before, the target users of GUIDE project are elderly people that possess accessibility issues, namely mild to moderate sensory, cognitive and physical impairments (loss or abnormality of psychological, physiological or anatomical structure or function) resulting from ageing or disability (restriction or lack of ability to perform an activity in the manner or within the range considered normal for a human being). In GUIDE, besides obvious forms of disabilities (e.g. locomotion difficulties, body disposition, carrying or moving objects) the mild forms of the following disabilities are also considered:

- Mental functions: mental functions include the functions of the brain and central nervous system, such as, consciousness, energy and drive, and specific mental functions, such as memory, language and calculation mental functions.
- Sensory functions: includes functions of sense (e.g. vision and audition).
- Neuromusculosketal and motor related functions: includes functions of movement and mobility, such as joints, bones, reflexes and muscles.

1.2.4 Input and Output Modalities

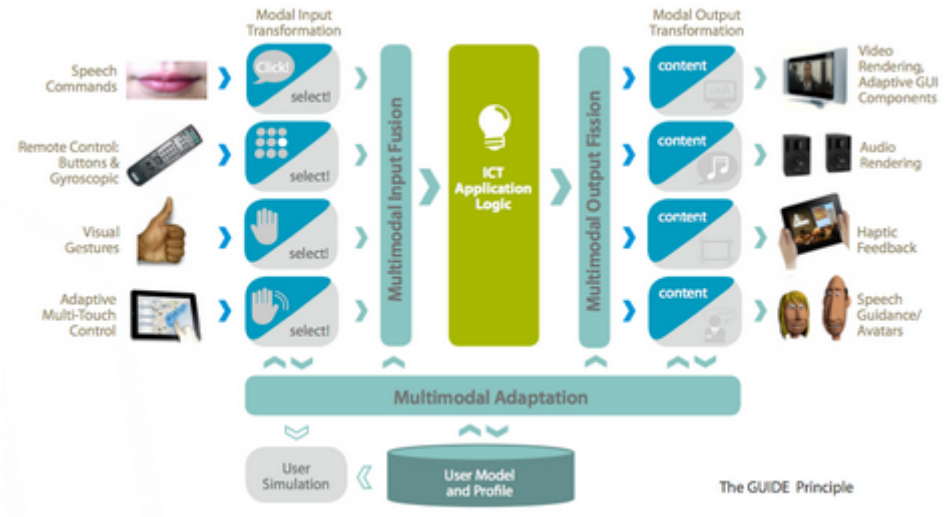


Figure 1.2: Input and output modalities/devices available in GUIDE

One of the purposes of project GUIDE is to make ICT (Information and communication technologies) applications accessible by providing user interfaces technologies that support traditional and novel interaction paradigms. Another goal is of course, to imprint adaptability into the framework, allowing it to choose the right interaction paradigm, faced with a certain user need or an application state.

Interaction paradigms or the modalities/devices considered for interaction purposes in GUIDE (see Figure 1.2) are briefly covered in the next subsections.

1.2.4.1 Visual human sensing and Gestures

Visual based human sensing usually intends to derive information from the actions of human beings, and those actions can either be explicit (e.g. gestures commands for controlling an application) or implicit (e.g. unintentional behaviour). In GUIDE scope, approaches based on infra-red and depth sensing for body tracking and gesture recognition, will be explored and evaluated to determine their applicability for multiple impaired users. Gestures can allow elderly people to have control over an application in a more natural and simpler way. In addition, deictic gestures to perform "pointing" tasks (e.g. selecting items on a programme list) as well as gestures on tablet PC devices to interact with the user interfaces will be used. Another very important aspect of visual human sensing in GUIDE, is the system task of performing person recognition. This process is truly critical, for instance, when dealing with multiple user's scenarios. In such situations, different user models/profiles are present in the TV STB, and so, the recognition process is useful for identifying which individual is interacting with the application and to select the correspondent profile. Not only is this useful in these situations, but also when occurs simultaneous use of the system by two or more users. By properly recognizing each one, the system can have a better grasp of all the interaction possibilities available and make adequate decisions such as presenting content over two modalities instead of just one, in order to cope with the user's accessibility requirements.

1.2.4.2 Audio

Interaction by means of audio, represent a major alternative to visual interaction for visually impaired users. It can be useful in a solo modality fashion by users whom have severe visual limitations or as a redundant/complementary modality for visual interaction, having greater or lesser relevance according to the situation or the characteristics of a specific person.

Since most elderly people today, are not used to traditional ways of interacting with user interfaces (e.g. keyboard, mouse), speech becomes a more natural and intuitive way to command a system. A modality like speech is even more suitable for the GUIDE project itself, because the usage of ICT is envisioned for private, indoor environments rather than public environment, which is prone to some issues such as ambient noise which may interfere with the recognition process. As for audio output, this can be of two forms, spoken language and abstract audio I/O.

1.2.4.3 Remote control and haptics

The current STBs and TVs provide their users traditional remote controls to easily interact with the hardware functionalities. This type of devices are cheap, extremely common and most of us are already familiar with them, however, using them can prove to be a difficult task for people with disabilities such as:

- Motor impairments - making it difficult to press tiny and/or close keys.
- Visual impairments - making it difficult to read the text on keys.
- Mild cognitive impairments - which makes the use of complex remote controls difficult due to the large number of keys and embedded features.

The approach to be used in GUIDE to resolve these issues is to make use of a gyroscope remote and haptic interfaces. This remote will mitigate motor and cognitive impairments effects, by including a very limited number of keys, being usable with little physical effort and at the same time providing an easy-to-understand natural interface similar to real objects (e.g. drag, select). The touch-sensitive feedback of the controller, limited number of buttons and the easy-to-read markings can aid visual impaired users.

1.2.4.4 Anthropomorphic user interfaces

Anthropomorphic user interfaces appear in GUIDE context as a set of embodied virtual agents (or avatars). Their use in an interactive system like this can bring several benefits; the dual utilization of audio combined with an avatar can increase the intelligibility of audio speech, therefore minimizing understanding effort; animated avatars can also easily attract attention, increase the impression of trustworthiness of a technical system [34, 37] and facilitate tasks completion by elderly users and users with cognitive impairments [36].

1.2.5 The Role of FCUL in GUIDE

An adaptable multimodal system is a complex system which need to possess several components with self-adaptation, including multimodal fusion and fission modules, as well as the main core part of the system, holding information about user context and history. Each one of these components plays a key role in a multimodal interactive system. A fusion module is responsible for the interpretation of the input from the user. That information would then be processed by the core of the system, according to applications logic. This system core usually handles the dialog management between components, holds data about users and environment, among other tasks. Finally, in order to compute and transmit the proper output feedback to users, a fission module determines the adequate message and way of displaying it. What was just described was the general structure and message flow in a regular multimodal system, ideas that will be further discussed along this thesis.

Our institution (FCUL) plays a major role in the GUIDE development process, being responsible for the overall multimodal adaptation features. The main task is to develop three of the core components in the GUIDE framework which are supposed to take advantage of adaptation to boost their performance in the user interaction experience: the multimodal fusion module, the dialog manager and the multimodal fission module. Adaptation is not only present in these components of the framework, but also ”outside” of it and therefore the role taken by FCUL extends beyond that, namely in the development of other pieces of software such as the User Trials Application and the User Initialization Application. Each one of these applications has a specific purpose in the project and a considerate part of this past year's work was devoted to them. They were important mainly for two reasons. Firstly they allowed an initial gathering of user requirements and preferences when interacting with a multimodal system such as GUIDE, and secondly it allowed us, the developers, to start grasping some of ”know-how” of designing and implementing a multimodal system as well as providing some ideas and concepts on how to implement the future modules we are responsible for. Two of the main chapters in this document were solely created to properly discuss both these applications and their contributions to the project.

1.2.6 My Role in GUIDE

In the beginning of the work related to this thesis, my main objective in the GUIDE project was to exclusively focus on the development of the multimodal fusion component. The initial plan was to study literature about the subject and implement earlier prototypes along with adaptation strategies. This process was supposed to conclude with the implementation of evaluation methods and a final integration with the rest of the components of the GUIDE framework. Due to project consortium decisions and delays, the goals of FCUL changed overtime, as discussed in the previous section, and consequently so did mine.

One of my goals in the GUIDE project, in conjunction with other colleagues, was to design and code the User Trials Application and the User Initialization Application. These two applications were considerably important in the first year of the project, because it was a year that has mainly concentrated on gathering and analysing user and stakeholders requirements. Aside from these two important tasks, the major objective and focus of this thesis work is on the development and integration of a self-adaptive fusion module in the GUIDE project. The adaptation of multimodal input has the goal of optimizing the parameters which control the integration of various information sources. On one hand, this information will be of direct nature, the input the system receives from the user directly, in the form of different modalities such as speech or gestures. On the other hand, secondary or inferred information will also be available and should be used to perform an efficient adaptivity, information which in the GUIDE project context, consists of user

profiles containing data about their capabilities, limitations and interaction patterns. The constant use and update of this "secondary" information becomes even more crucial when dealing with elderly people because of their special needs and particularities involving physical and cognitive impairments.

1.3 Results

The following are the main results of this thesis:

- The User Trials Application, an application that was created and updated during the first year of GUIDE, and that was of extreme importance to perform live user tests and gather a large quantity of data about user requirements and preferences.
- The User Initialization Application, an application that was created as a prototype for the final version that will be running inside the GUIDE framework and which demonstrates the concept of input combinations and dynamic adaptation.
- The Multimodal Fusion Module, a core component of the GUIDE framework, capable of receiving user input through various modalities (e.g. key presses on remote control, speech) and combine that information according to the current state of an application. This component will, in the future, go through further iterations of development to support all of the project needs.

1.4 Contributions

The work developed during the course of this thesis gave birth to some publications which are the following:

Support for inferring user abilities for multimodal applications
Carlos Duarte, Pedro Feiteira, David Costa, Daniel Costa
in Proceedings of the 4th Conferência Nacional em Interacção Pessoa-Máquina (Interacção 2010), Aveiro, Portugal, 2010

This paper presented a "Wizard-of-Oz" approach, intended to strengthen a multimodal application with the objective of defining a user. This allows a "Wizard" to replace some components of the system (e.g. input recognizers), while still supporting the goal of assisting in understanding which user characteristics are relevant for an application in development and how different users combine different modalities. A similar approach was taken when developing one of the applications described in this thesis, the User Trials Application.

Eliciting Interaction Requirements for Adaptive Multimodal TV based Applications
Carlos Duarte, José Coelho, Pedro Feiteira, David Costa, Daniel Costa
in Proceedings of the 14th International Conference on Human-Computer Interaction (HCII), Orlando, Florida, USA, 2011

A user-centred approach to elicit interaction requirements, understand how user's abilities impact their perception and how they use use their skills, is the main topic of this paper. The results presented several observation of user interaction and empowered the necessity of having adaptive behaviours to deal with a user population with a broad diversity of skills.

Adaptive Multimodal Fusion
Pedro Feiteira, Carlos Duarte
in Proceedings of the 14th International Conference on Human-Computer Interaction (HCII), Orlando, Florida, USA, 2011

This paper focuses in multimodal fusion, a critical concept that is essential in any multimodal system. An overview of the state-of-the-art is given, including architectural approaches, adaptation and benchmarking. Many of the ideas discussed were the basis for the development of the GUIDE fusion core, one of the major components of the GUIDE framework that is addressed during this thesis.

1.5 Document structure

This introductory section has served the purpose of giving the reader a brief insight on the importance of adaptivity in multimodal systems and how elderly or impaired user can benefit from it. An explanation of GUIDE followed, explaining how application developers can benefit from the toolbox the project intends to deliver and the specific role of this thesis work, along with a description of main objectives. The remainder of the document is structured as follows:

- Chapter 2 covers related work associated with this thesis. It begins by granting the reader with some thoughts on multimodal interfaces and their features. Then a focus is given to a particular part of a typical multimodal system, the multimodal fusion, responsible for input combination and that can possess different architectural and algorithm approaches. The adaptivity property of fusion is also addressed and constitutes a major concept throughout this document. Fusion engines are fur-

ther discussed for the rest of the chapter, including historical backgrounds or novel topics in the area such as benchmarking and evaluation.

- Chapter 3 introduces the User Trials Application. It states why such an application needed to be developed and its importance on the overall project development. An overview of the architecture is given as well as the type of tests implemented and how they can be constructed and run. The chapter finalizes by focusing on the user trials performed, including the set-up, user tasks and a brief discussion on results that are relevant for the process of multimodal fusion.

- Chapter 4 describes the User Initialization Application and the respective prototype that was developed. It starts by explaining what the UIA is, and why is such an application a critical part of the GUIDE framework. An architectural point of view is given, not only to describe the components and their interactions but also to emphasize this application self-adaptation feature. The chapter concludes by giving an insight on the type of tests implemented.

- Chapter 5 covers the GUIDE Fusion Core, one of major components inside the GUIDE framework. After an introduction to what is the GFC and its importance to the framework, a description of the architecture is given, along with the events that are exchanged and how it interacts with other components. The specific approach that allows the combination of inputs is discussed and afterwards an explanation of how adaptive behaviour is imprinted into that approach follows. The evaluation framework and methods used to evaluate the implemented fusion engine is discussed in the end of the chapter.

- To conclude this thesis chapter 6 will provide some conclusions about the work developed during the past year and give some insights on the future work that is still to be done regarding the GUIDE fusion core.

Chapter 2

Related Work

The second chapter of this report will focus on the concepts already defined by scientific contributions on multimodal interfaces (Section 2.1) and multimodal fusion (Section 2.2). Finally, after the unveiling of basic ideas and concepts, Section 2.3 presents the work developed in the field of fusion engines across the years, giving some insight on some applications already developed and the future endeavors that must be taken and considered.

2.1 Multimodal Interfaces

In our everyday lives we are constantly communicating with each other, by means of diverse modalities like speech, gestures or vision. Thus, almost any natural communication among humans involves multiple, concurrent modes of communication [42]. Considering this, we can safely state that multimodal interaction is regularly present in ordinary human-to-human dialog. This sort of communication is also often desirable in human-machine interaction, but fundamental changes haven't been observed in the couple of decades [35] following Richard Bolt's work with the "Put-That-There" paradigm [2]. However, multimodal interaction has shown much development in the past decade [4]. On one hand it provides a more "human" way of interacting with computers, by means of gestures, haptic, speech, or other modalities, as well as being the primary choice over unimodal interfaces by users [43]. On the other hand this type of interaction has also demonstrated to offer better flexibility and reliability than any other human/machine interaction means [44].

In the following subsections some concepts involving multimodal interfaces will be further explained and discussed, mainly the aims of its research, features and advantages.

2.1.1 Aims

In the beginning of HCI history, the standard way for a human to interact with a computer was through deterministic and well defined WIMP(Window, Icon, Menu, Pointing device)

interfaces, which relies in a single mode of interaction, providing input through rudimentary devices like the mouse, keyboard or joystick. This form of interaction may prove useful and more efficient in some cases, but generally speaking, HCI can become richer if multimodalities are employed in the interaction process, giving a wider range of choice to the users [42]. As stated by Oviatt [4], « Multimodal interfaces process two or more combined user input modes (such as speech, pen, touch, manual gesture, gaze, and head and body movements) in a coordinated manner with multimedia system output. They are a new class of interfaces that aim to recognize naturally occurring forms of human language and behavior, and which incorporate one or more recognition- based technologies (e.g. speech, pen, vision) ». Thus, it's safe to assume that a computer system that wants to comprehend human language in a so called "natural way" must be multimodal. Oviatt et al. [43] also distinguish multimodal architectures and processing because they possess two particular characteristics: fusion of different data types, or also called information fusion [45]; and real-time processing and temporal constraints imposed on information processing. This definition of multimodality is also called or viewed as a system-centered definition [26], while the user-centered definition consists on the use of different modalities (referring to the human senses) [26].

2.1.2 Features

Unlike other type of human-machine interaction, multimodal interfaces aim to provide a more transparent and natural way of interaction to users, making use of several modalities like speech, gestures, gaze, etc. With all these interaction possibilities at their disposal, it is expected from this type of interfaces to be able to enhance human-computer interaction in a number of ways [4] including:

- Enhanced robustness due to combining different partial information sources;
- Flexible personalization based on user and context;
- New functionality involving multi-user and mobile interaction.

Of all these features, the first one is particularly interesting on this thesis scope. That's because it relates to the concept of fusion, a process capable of extracting meaning from collected information. Sometimes the data received from a single input, may not be enough to extract meaning from it, and so, by having more information sources, they can complement each other. An example of this kind of situation would be a crowded environment where speech recognition can't function properly due to loud environment noise. By making use of another modality, like lips movement recognition, both information sources could be combined to produce reliable results. A multimodal system strives for meaning [26]. Multimodality in an interactive system isn't only about the use of many

GUI	MUI
Single input stream	Multiple input streams
Atomic, deterministic	Continuous, probabilistic
Sequential processing	Parallel processing
Centralized architectures	Distributed and time-sensitive architectures

Table 2.1: Main differences between GUI and MUI according to Oviatt et al. [30]

modalities; it is also about extraction of meaning from all the possible actions towards that same system.

Oviatt et al.[30] when comparing multimodal interfaces to the standard GUI interfaces, have drawn the differences expressed in Table 2.1. In a standard GUI interface, a single input is used to interact at any given time, be it the mouse, keyboard or any other device. With multimodalities this isn't the case, since many inputs are captured through many recognizers and its information processed. Obtaining this data is a continuous process that involves interpretation from probabilistic recognizers, being these results weighted by a certain degree of uncertainty. Input streams of GUIs on other hand, are generally deterministic, with key strokes and mouse position controlling the computer. Because of this characteristic the processing is pretty much sequential. Multimodal systems have a lot of recognizers to obtain input, thus it's obvious one of their properties is time synchronized parallel processing, assured by time- sensitive and distributed architectures, in order to deal with synchronization and computation needs.

2.1.3 Advantages

The interaction of humans with the environment, including other humans, is by nature, multimodal. We talk to people and point at the same time; we hear what someone says and observe their facial expressions to comprehend their present emotions. In the HCI world however, the user is usually confronted with the display of a single screen, where unimodal interaction takes place, resulting in a bigger effort by the user to effectively express his intent to the computer. Here are some advantages or practical reasons why one should make use of multimodal systems in HCI interaction.

2.1.3.1 User satisfaction

One of the main reasons why people use multimodal systems is because they like it. "Stone-age" devices like the mouse, joystick, or keyboard, limit the ease with which a user can interact in today's computing environments, including, for example, immersive virtual environments [42].

Several studies based on the Wizard-of-Oz approach (the role of the multimodal system is played out by a human) were made, which revealed that people favor multiple-action modalities for virtual object manipulation tasks [13, 29]. Oviatt [28] has also shown

that about 95% of users prefer multimodal interaction over unimodal interaction, using, for example, gestures together with speech. The existence of redundant input in these kinds of systems is also a plus for cognitively and physically impaired people. Indeed the abundant quantity of information originated from the various actions users can do (due to the many recognizers present in such systems to capture different modalities) allow them to choose whichever type of modality suits them better.

2.1.3.2 Robustness and Accuracy

Single modality HCI suffers from lack of robustness and that is because some technologies, like speech recognition for example, are highly susceptible to noise or information loss. Multimodal interaction mitigates these disadvantages, because by combining more than one source of information, decisions can become more robust and reliable. A good example of this robustness is when a user is waving unintentionally, and that gesture is incorrectly interpreted by the recognizer as a command. In this case a combination of gestures and speech would work better to understand that the user's action really meant intent towards the system.

2.1.3.3 Efficiency and Reliability

Multimodal interfaces were first seen as more efficient than unimodal interfaces, and later on some evaluations came to prove that this was true, they could in fact speed up tasks completion by 10% [28]. Another characteristic of these interfaces and perhaps more worth noting and astonishing at first sight, is that multimodal interfaces have been shown to improve error handling and reliability: users made 36% fewer errors with a multimodal interface than with a unimodal interface [28]. Because of their lack of determinism and abundance of available options to interact, one could think that the interaction process would be more error and failure prone in multimodal interfaces, but it's been demonstrated to not be the case.

2.1.3.4 Adaptivity

Multimodal interfaces should and can adapt to the needs and abilities of different users, as well as different contexts of use. As stated in section 2.1.3.1, users with physically or cognitive limitation can benefit from this advantage. Data about users and their individual characteristics (age, sensory or motor impairment) can be represented by a user profile, which will be used by the system in order to enable a dynamic adaptation of the interface screens.

2.1.4 Architecture and Key Components

In this section, it will be described from a top-level view, the major software components that a multimodal system should contain. Various terms have been widely accepted, like fusion engine, fission module, dialog manager and a context manager, which all together form what is called the "integration committee" [11].

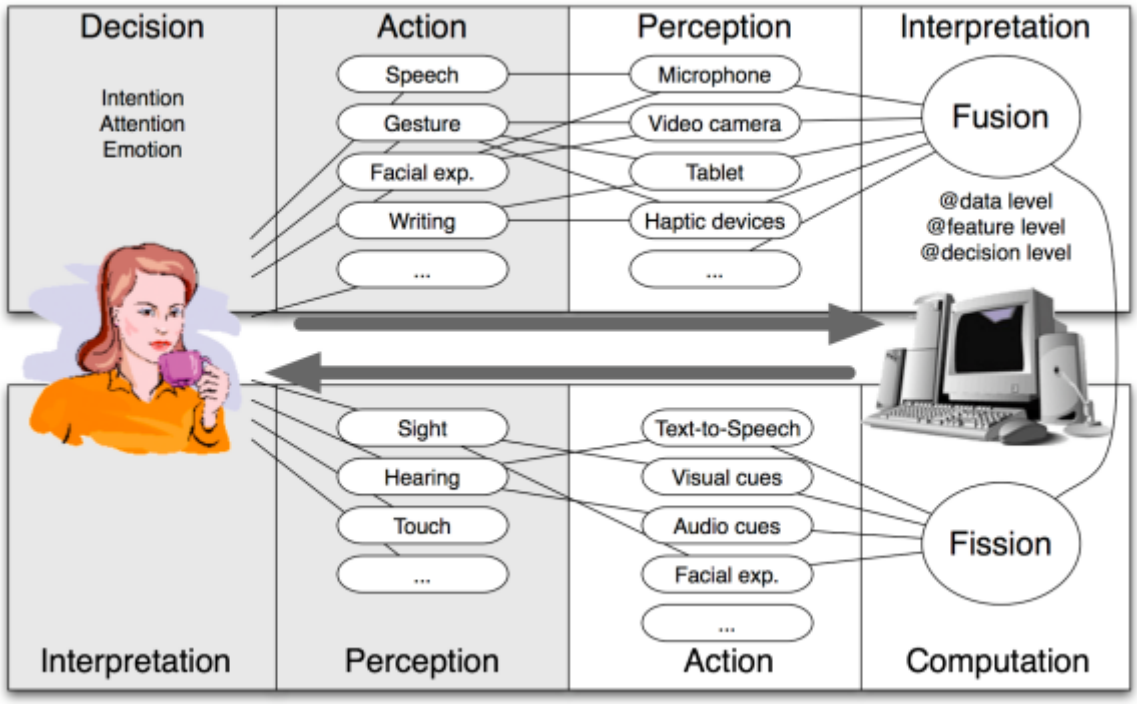


Figure 2.1: The 4 states of user and machine in multimodal interaction according to Dumas et al.[4]

Figure 2.1 is based on the conclusions in [4] about the general flow of communication between user and machine, which occurs in the multimodal interaction process. As we can see, a human has four main states when interacting with a multimodal system. At first, there's the decision state, when conscious or unconscious intent is formulated in the user's head. In the next state, the action state, the message processed on the previous state, will have to be passed to the system, so the user picks the communicating means of their choice and execute it, using gestures, speech, or others ways of expression. Similarly to the user, the computer also has four states. In the perception state, individual recognizers or sensors work to assimilate all the information coming from the user. Then, that information will be analyzed by the machine in the interpretation state, trying to extract meaning from all the data collected. This process is also known as information fusion, and can be done at different levels, as it will be explained further in this report. After this analysis, the computer understands what has to be done, and it's ready to start preparing its response. The computation state is when the fission module starts to work, making decisions about how to present the adequate feedback to the user, following the business

logic and dialog manager rules defined by the developers. After this computation phase, an answer is obtained and the system goes into the action state, transmitting that answer to the user through the available and proper tools (text-to-speech, visual cues, etc.). When the user perceives this message, he enters the perception state, followed finally by the interpretation state, which will allow the user to make sense of the information displayed to him.

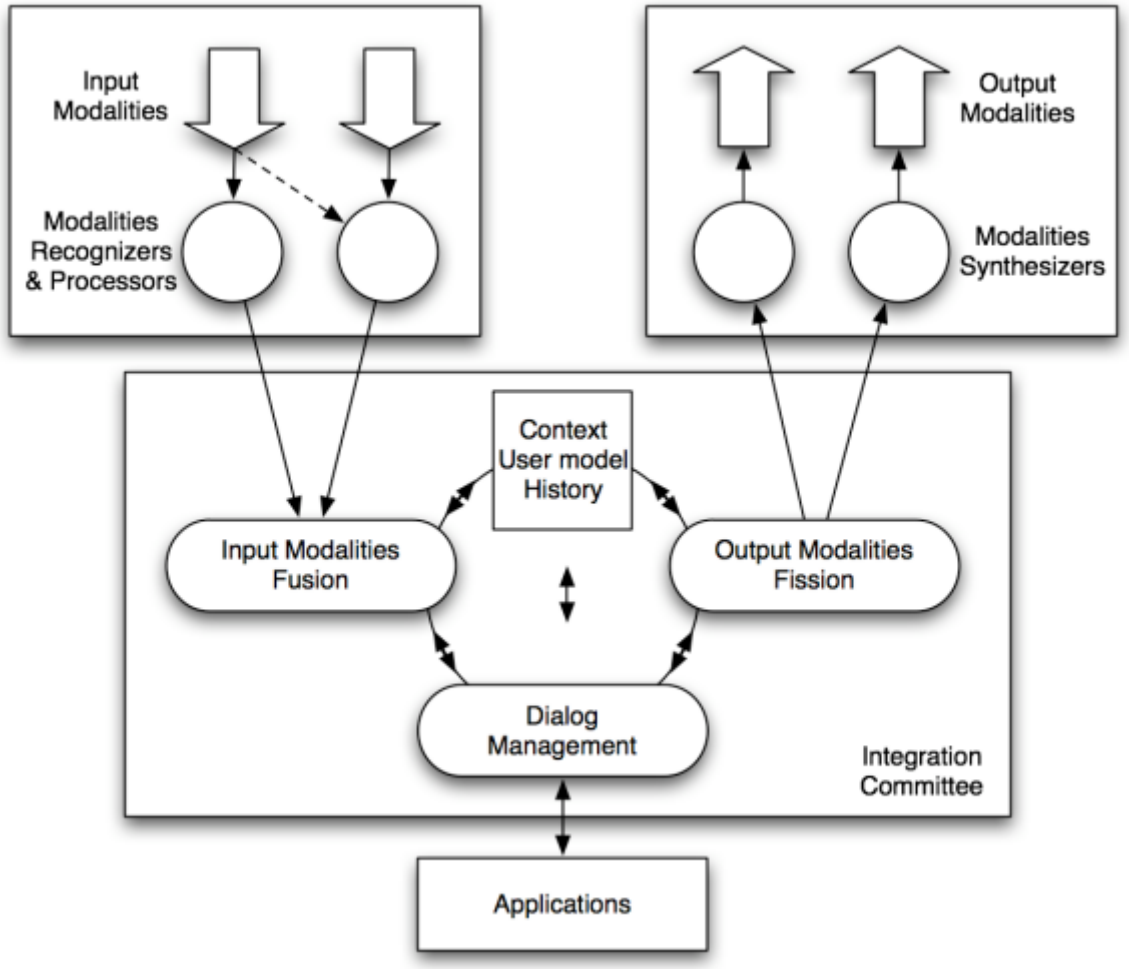


Figure 2.2: The general architecture for multimodal systems according to Dumas et al.[4]

In figure 2.2 we can observe the general architecture of a multimodal system, along with its major software components. This figure demonstrates on a software vision, the message flow in multimodal systems, from the user to system, even including the developer applications. As we can see, input modalities are first perceived through various recognizers, which output their results to the fusion engine, in charge of giving a common interpretation of the inputs. When the fusion engine comes to an interpretation, it communicates it to the dialog manager, in charge of identifying the dialog state, the transition to perform, the action to communicate to a given application, and/or the message to return through the fission component. Finally, the fission engine is in charge of returning

a message to the user through the most adequate modality or combination of modalities, depending on the user profile and context of use. For this reason, the context manager, in charge of tracking the location, context and user profile, closely communicates any changes in the environment to the three other components, so that they can adapt their interpretations.

2.2 Multimodal Fusion

In multimodal interactive systems, multimodal fusion is a crucial step in combining and interpreting the various input modalities, and it's one of the distinguishing features that separate multimodal interfaces from unimodal interfaces. The aim of sensor fusion is to analyze many measurements simultaneous, and try to construct semantic meaning from them, which would be harder if only individual measurements were taken into account. When meaning is extracted from all the information collected, that message will then be passed to a human-machine dialog manager, in charge of managing the communication between the system software components.

Nigay & Coutaz [26] published one of the first scientific papers involving fusion in multimodal interfaces, and defined how they can handle inputs in different ways in order to make sense of a set of information provided by the various modalities.

		USE OF MODALITIES	
		Sequential	**Parallel**
FUSION	**Combined**	ALTERNATE	SYNERGISTIC
	Independent	EXCLUSIVE	CONCURRENT

Table 2.2: Ways to interact with multimodal interfaces. Two dimension from the classification space presented by Nigay & Coutaz [26]

The "Use of modalities" columns of Table 2.2, expresses the temporal availability of modalities, while the lines represent the fact that information obtained from several modalities can be either combined or treated in an independent fashion. While sequential use of modalities forces the user to use them one at a time, the support for "parallel" use of modalities, allows the user to employ multiple modalities at once, increasing the rate of information transmission between user and system. If this information is further combined, it becomes a synergistic form of interaction [26].

The next subsections are organized as follows: Section 2.2.1 will present the existent levels at which fusion of different modalities can be executed. Section 2.2.2 will present the possible architectures for decision-level fusion, the most common type of fusion, and finally section 2.2.3 will introduce some ideas about adaptive possibilities in multimodal fusion.

2.2.1 Levels of Fusion

Based on the type of information available, different levels of fusion may be defined and used. Sharma et al. [42], considers three levels for fusion of incoming data: sensor-level (or data-level) fusion, feature level-fusion and decision level-fusion. Sanderson and Paliwal [8] however, found more intuitive to categorize the various levels of fusion into three main categories: pre-mapping fusion, midst-mapping fusion and post-mapping fusion (See Figure 2.3). In this context, the pre-mapping consists on performing information combination before any use of recognizers or experts is made. If a midst-mapping approach is used then the information is to be combined while a mapping from sensor-data/feature space into opinion/decision space occurs. With post-mapping fusion on the other hand, this space mapping operation produced by classifiers/experts is executed before the information fusion take place. Sanderson and Paliwal [8] distinguish classifier and expert; while the former provides a hard decision, the latter provides an opinion (e.g. in the [0,1] interval) on each possible decision.

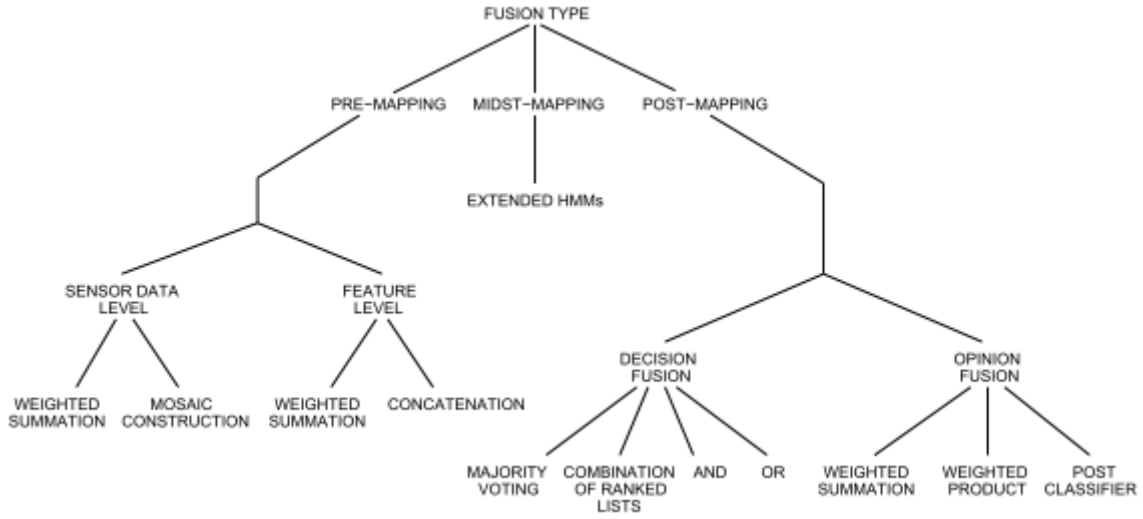


Figure 2.3: Different levels of information fusion (adapted from Sanderson and Paliwal [8]

As stated before, the "fusion after mapping" approach consists on making use of information that already was pre-processed by invoked matchers. This way the amount of information will be greatly reduced, allowing an easier analysis and response. Pre- classification fusion schemes typically require the development of new matching techniques (since the matchers/classifiers used by the individual sources may no longer be relevant) thereby introducing additional challenges [38]. The following subsections will explain with more detail the types of fusion existent, focusing on the classification pointed out by Sanderson and Paliwal [8]. The tree hierarchy in figure 2.3 illustrates the approaches which will be now discussed and table 2.3 summarizes their main characteristics.

2.2.1.1 Pre-mapping: Sensor Data-level fusion

Data-level fusion, also called sensor-level fusion, deals with raw data coming from the recognizers, representing the richest form of information possible (quantitatively speaking). Because the signal is directly processed, no information loss occurs. It's normally used when dealing with multiple signals of the same type, involving one modality only. Depending on the application involved, there are two methods to accomplish this type of fusion[8]: weighted summation and mosaic construction. Weighted summation can be used for instance in combining the information of a regular and an infra-red image to produce a new one. One example of mosaic construction is when two or more video cameras capture different point of views of a scene in order to create a new image. The down-side of data-level fusion is its susceptibility to be affected by noise and failure.

2.2.1.2 Pre-mapping: Feature-level Fusion

Feature-level fusion, is a type of fusion oriented for closely-coupled or time synchronized modalities such as, for example, speech and lips movement recognition. In this type of fusion, features are extracted from data collected by several sensors. If the features are commensurate they can be combined by weighted summation (e.g. features extracted from data provided by two microphones). If they are not commensurate then feature vector concatenation can be employed, where a new feature vector can be constructed by concatenating two or more feature vectors (e.g. to combine visual and audio features) [8]. One of the disadvantages of the features vector method is that separate feature vectors must be available at the same frame rate [8]. Performing a synchronous feature extraction can prove difficult when dealing with, for instance, visual and audio features, because of their usual different rates of extraction.

Unlike data-level fusion, it can suffer from data loss, but manages noise interference better. The most classic architectures used for this type of fusion are adaptive systems like artificial neural networks, Gaussian mixture models, or hidden Markov models [4]. The use of these types of adaptive architecture also means that feature-level fusion systems need numerous data training sets before they can achieve satisfactory performance [4].

2.2.1.3 Midst-Mapping Fusion

In midst-mapping fusion several information streams are processed concurrently while the mapping sensor-date/feature space to decision/opinion space takes place. This type of fusion, similarly to feature-level fusion, is also oriented for closely coupled modalities such as lips and speech recognition. However, it avoids some problems like the need to match frame rates (See section 2.2.1.2) [8]. Extended Hidden Markov Models are an example of an architecture possible for this kind of fusion [8].

2.2.1.4 Post-Mapping: Decision-level Fusion

One of the most common and widely accepted forms of fusion is decision-level fusion, and that is because it allows multimodal systems to make effective use of loosely- coupled modalities, like speech and pen interaction. Because the information received by the fusion engines has already been processed, noise and failure are no longer issues to deal with. This means, that fusion will have to rely on preprocessed information in order to construct semantic meaning from combining partial semantic information coming from each input mode. That preprocessed information constitutes a hard decision that was produced by one or more classifiers. According to Sanderson and Paliwal [8], those decisions can be combined (to construct the mentioned semantic meaning) through several different approaches:

- Majority Voting - majority voting is a pretty much straightforward technique to reach one final decision. In this approach a consensus is reached on the decision by having a majority of the classifiers declaring the same decision [8].

- Ranked List Combination - In HCI applications, the output of the system can be viewed as a ranking of plausible hypotheses. In other words, the output indicates the set of possible hypotheses sorted in decreasing order of confidence. The goal of rank level fusion schemes is to consolidate the ranks output by the individual recognizers systems in order to derive a consensus rank for each hypothesis. The ranked lists can be further combined taking in account, for example, the reliability and discrimination ability of each classifier [8]. The usual way of selecting one final decision, is to select the one on the top of the list.

- AND Fusion - In AND fusion a final decision is reached if and only if all classifiers agree. This approach can be really restrictive, because if a lot of classifiers are involved in the decision making process, a final decision may never be reached. A good example of a system where this type of fusion could prove useful is biometric recognition systems where tolerance for failures is minimal or non-existent.

- OR Fusion - In OR fusion a final decision is reached as soon as one of the classifiers decides something. Unlike AND fusion, this approach is very relaxed and one scenario can present several different decisions to its problems.

2.2.1.5 Post-Mapping: Opinion-level Fusion

Opinion-level fusion (also called score-level fusion) is very similar to decision-level fusion because both of them operate after the mapping of data/feature-level space into decision/opinion space. In fact, some literature (e.g. [23]) considered the former as a sub-set of the latter. However, in the case of opinion-level fusion, a group of experts

provides opinions instead of hard decisions, and for that reason Sanderson and Paliwal [8] found more adequate to make a distinction between the two types. When observing ranked-list combination fusion the ranks on the list itself could be considered to indicate an opinion from the classifier. However, there are two differences between a ranked list and a match/score-level approach. While the former provides more insight into the decision-making process of the matching compared to just the best hypothesis, it reveals less information than the latter [38]. However, unlike rank-level fusion, usually outputs from score-level fusion are not directly comparable (because heterogeneous experts can be used), so they have to be commensurate before any further processing (e.g. by mapping the output of each expert to the [0,1] interval, where 0 indicates the lowest opinion and 1 the highest opinion [8]). Because of this necessity, rank-level fusion schemes are simpler to implement compared to the score-level fusion techniques [22]. Opinions combination can be achieved, for example, through weighted summation or weighted product approaches (both briefly covered in section 2.2.3), before using a classification criterion (e.g. MAX operator) in order to reach a final decision. The main advantage of these approaches over feature vectors concatenation and decision fusion is that opinions from each expert can be weighted [8]. This allows to imprint adaptive features into a system, by setting the reliability and discrimination of experts through time according to the state of the environment/signal quality, users, or application logic (see section 2.2.3).

2.2.2 Architectures

Since decision-level and opinion-level fusion are the most common types of fusion used today, because of its flexibility on dealing with loosely-coupled modalities, only this kind of fusion will be considered as usable on this thesis project work. Some architectural approaches to both decision and opinion-level fusion were explained in the previous section. In addition, Dumas et al [4] consider the following as typical choices to decision-level architectures:

- Frame-based fusion: uses data structures called frames or features for meaning representation of data coming from various sources or modalities. These structures represent objects as attribute-value pairs.

- Unification-based fusion: based on recursively merging attribute-value structures tries to obtain a logical whole meaning representation.

- Symbolic/statistical fusion: an evolution of standard symbolic unification- based approaches, which adds statistical processing techniques to the frame-based and unification-based fusion techniques. This kind of fusion is also called “hybrid” and has demonstrated to be able to achieve robust and reliable results [4]. Examples

	Data-level fusion	Feature-level fusion	Decision-level fusion	Opinion-level fusion
Input type	Raw data of the same type	Closely coupled modalities	Loosely coupled modalities	Loosely coupled modalities
Level of information	Highest level of detail	Moderate level of detail	Disambiguation by combining data from different modules	High level information used for constant adaptation
Noise/failures sensitivity	Highly susceptible to noise or failure	Moderately susceptible to noise or failure	Highly resistant to noise or failure	Highly resistant to noise or failure
Usage	Not really used to combine modalities	Used for fusion of particular modes	One of the most widely used types of fusion	One of the most widely used types of fusion
Application examples	Fusion of two video streams	Speech recognition through voice and lips movement	Pen/speech interaction	Pen/speech interaction

Table 2.3: Main differences between data-level, feature-level, decision-level and opinion-level fusion adapted from [4]

of the usage of hybrid fusion schemes can be seen in the MTC(Member-Team-Committee) architecture used in Quickset [31] and in the work of Hong and Jain [24], which used both a fingerprint expert and a frontal face expert in a fusion scheme involving a ranked list and opinion fusion.

2.2.3 Adaptive Fusion

Fusion classifiers can be distinguished not only by the type of fusion or architecture they possess, but also by whether they are adaptive or non-adaptive [38]. The basic concept around adaptive fusion (also called quality fusion) is to assign different weight values associated with a modality. As stated in section 2.2.1.5 Sanderson and Paliwal [8] pointed out two examples of how such weighting can be used in performing an adaptive opinion fusion; weighted summation fusion and weighted product fusion, which will be briefly discussed in sections 2.2.3.1 and 2.2.3.2 respectively. Section 2.2.3.3 will present a similar yet different perspective on adaptive vs. non adaptive fusion by Poh et al. [38].

2.2.3.1 Weighted summation fusion

In weighted summation the opinions regarding class j (in this case a class can be viewed as an input-related event observed by the system) collected from N_E experts can be combined using:

$$f_j = \sum_{i=1}^{N_E} w_i o_{i,j}$$

In this formula $o_{i,j}$ stands for the opinion of the i-th expert and w_i represents the weight associated with each expert. The weight used for an expert can vary depending on many factors (e.g. decrease of an audio expert weight can happen in a situation with low audio speech noise ratio conditions) and its value is set in a [0,1] interval, with the constraint $\sum_{i=1}^{N_E} w_i = 1$. This approach is also known as *linear opinion pool* [15] and *sum rule* [1, 21].

2.2.3.2 Weighted product fusion

Assuming that experts are independent, the opinions regarding class j collected from N_E experts can be combined using:

$$f_j = \prod_{i=1}^{N_E} o_{i,j}$$

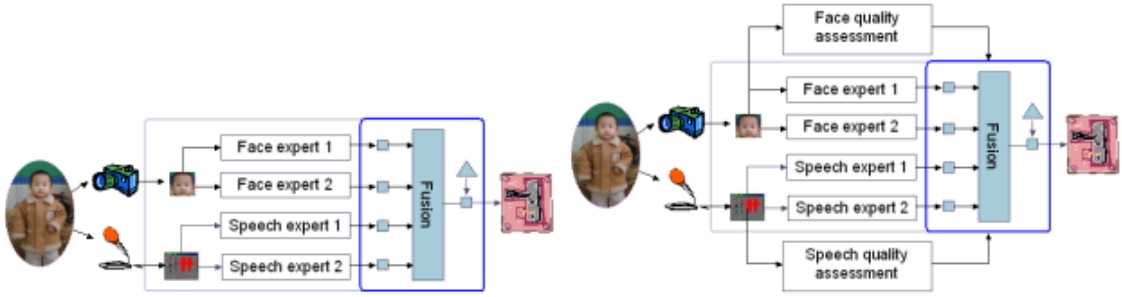
In order to take into account discrimination and reliability of each expert, weighting can be introduced in the formula above in the following manner:

$$f_j = \prod_{i=1}^{N_E} (o_{i,j})^{w_i}$$

This weighted product approach is also known as *logarithmic opinion pool* [15] and *product rule* [1, 21]. According to Sanderson & Paliwal [8] weighted product fusion presents two disadvantages; First of all, when an opinion from one single expert is assigned a near to zero value, the overall result will also near to 0, which makes each expert have a great influence in the combination process outcome. The second pointed downside is that the expert independence only holds true when each one is using independent features (e.g. audio and video).

2.2.4 Adaptive fusion through signal quality

Poh et al [38] state that adaptivity work as a function of the signal quality measured on one modality. The idea is, the higher quality a signal has, more weight will be set for it. One use of this kind of adaptation is for instance, a person's recognition in a biometric system. Because the light conditions can change and influence the system input (in this case, the



(a) Non-adaptive fusion (b) adaptive fusion with quality

Figure 2.4: Adaptive vs non-adaptive fusion according to Poh & Kittler [38]

face recognition), this visual modality may get a lower weight value whilst speech input would get a higher value, and thus considered more trustworthy in the recognition process.

According to Poh & Kittler [38], signal quality can be measured through quality measures. These measures are a set of criteria used to assess the incoming signal quality of a modality. Such measures, could be for example, lighting or reflections in face detection and SNR(speech noise ratio) for sound. An ideal quality measure should correlate, to some extent, with the performance of the classifier processing the modality [38]. This means that some characteristics of a classifier prone to affect its performance should make ideal quality measures (e.g. if head pose in face recognition affects the recognition process then head pose would serve as an optimal quality measure). Figure 2.4 represents the difference between adaptive and a non-adaptive approach stated by Poh & Kittler [38].

2.3 Fusion Engines

The concepts and ideas brought to life by multimodal fusion are materialized in the form of fusion engine, a core component in any adaptive multimodal system. The following subsections will talk about some of the work developed and to be developed regarding fusion engines.

2.3.1 Historical perspective

As mentioned in section 2.1, since Richard's Bolt presented his paper about the work developed in the "Put-that-there" interactive system, thirteen years have since passed without any significant contributions to the multimodal interfaces field of study. In 1993, Nigay & Coutaz [26] presented a paper tackling the analysis of the integration of multiple communication modalities within an interactive system, triggering a new wave of scientific studies focused on fusion engines. In this historical perspective, it will be presented the previous mentioned scientific contributions and evolution of fusion engines along the years. To that end, it will be used Brian Gaines's model of technological development

and diffusion [41]. This model goes by the name of BRETAM, which represents all the 6 stages it includes: **B**reakthrough, **R**eplication, **E**mpiricism, **T**heory, **A**utomation and **M**aturity. The next subsections will be about each one of these stages relating to the fusion engines context, based on the vision of Lalanne et al. [11].

2.3.1.1 Breakthrough Phase

According to Gaines's framework, every technology begins with a breakthrough, a period when the very first concept/idea comes to life. In the field of fusion engines this has happened with Bolt's "Put-that-there" paradigm [2]. This work already made use of multiple modalities, more specifically speech and pointing- based gestures. One could create and move objects in a 2D space by pointing at them and using spoken commands. Although combination of information was in fact happening in this system, the concept of fusion engine was neither introduced nor discussed at that time.

2.3.1.2 Replication Phase

Further researched allowed the transition of the research field from the Breakthrough phase to the Replication phase. The work in this phase, as far as fusion engines are concerned, has identified some issues to consider, however these remained in a high level of abstraction, setting a primary focus on problems identification rather than proposing solutions. During this period there were two major contributions:

- CUBRICON [14] that uses speech with deictic gestures and graphical expressions in a map application. The system combines the input streams into a single compound stream having temporal order of the tokens. The parser corresponds to a state-based model represented by a generalized augmented transition network. CUBRICON contains a set of rules for inferring the intended referent in case of ambiguity. This is done by either selecting the closest object that satisfies the criteria or by issuing an advisory statement describing the inconsistency. These disambiguation rules (in addition to the input stream fusion) can be considered as the first explicit representation of fusion engine behavior. [11]

- Xtra [47] (eXpert TRAnslator) which is an interactive multimodal system based on keyboard for Natural Language and mouse pointing as input modalities. The underlying idea of Xtra is to exploit a multi-modal interfaces in order to increase the bandwidth between the user and the underlying tax declaration system. [11] The implementation of Xtra described in [47] provided natural language access to an expert system, capable of assisting the user in filling out a tax form. This system brought up some interaction issues to consider such as the integration of more complex pointing actions like a continuous finger movement tracking or handling pointing in a 3-D space.

These two contributions were in fact, the very first real engineering work and steps on fusion engines development. However CUBICRON and Xtra only deal with modalities in a sequential way (See Table 2.2), making them somewhat limited.

2.3.1.3 Empiricism Phase

When the Empiricism Phase takes place, several experiences had already occurred and therefore valuable lessons and experience can be drawn from them. From that knowledge it's then possible to formulate empirical design rules that prove useful in further work.

Lalanne et al. [11], considers that this phase is composed by four major contributions: the integration of speech, gaze and hand gestures by Koons et al. [7], the PAC-Amodeus architecture and its fusion engine [27], the Quickset platform [40] and the fusion engine by Johnston and Bangalore [33].

Koons et al [7] studied three modalities: gaze, speech and hand gestures in the "blocks world", a graphical 3D system. Modalities are first parsed individually; the parsers then produce the information in a common frame-based format for fusion. All the information is received in parallel and is time-stamped. [11]

Pac-Amodeus [27] is a software architecture model that makes use of a generic fusion mechanism for designing and implementing multimodal interaction. Since the Pac-Amodeus along with its fusion engine, constitute a global platform applicable to the software design and implementation of multimodal interactive systems, we can see it as a perfect fit in the empiricism phase. The architecture is illustrated by making use of MATIS (Multimodal Airline Travel Information System), which allows a user to retrieve information about flight schedules using speech, direct manipulation, keyboard and mouse, or a combination of these techniques. The Pac-Amodeus architecture possesses a component named "Dialog Controller", where data fusion takes place at a high level of abstraction by PAC agents, using a common representation, a Melting Pot. This uniform format is a 2D structure representing an event with structural and temporal information. The criteria for triggering fusion are threefold: the complementarity of melting pots, near time, and context rules. When triggered, the engine attempts three types of fusion in the following order: microtemporal fusion, macrotemporal fusion, and contextual fusion [27]. Microtemporal fusion is performed if the structural parts of the melting pots are complementary and if their time interval overlaps. Macrotemporal fusion is used if informational units were processed sequentially by the user or if the system couldn't process them in a parallel way and had to compute them in sequence instead (for example due to lack of computational resources). This kind of fusion is thus adequate when having complementary structural parts of melting pots not overlapping in time but belonging to the same temporal window. Contextual fusion on the other hand disregards attention for temporal constraints and it's driven by the current active context.

Quickset [40] is a collaborative, handheld, multimodal system for interacting with dis-

tributed applications. This interactive system features the use of two modalities: graphical (using pen-based interaction) and speech (using a voice recognition system). These modalities are used to control Leathernet which is a simulation system for training of US Marine Corps platoon leaders. Quickset has also been used with MIMI a search engine for finding health care facilities. Quickset makes use of a unification-based mechanism to perform fusion of partial meaning representation fragments derived from the input modalities. If that fragments prove to be "compatible" they will then be fused into a single result. Fusion is done through constant analysis of the two event streams to determine potential integration, the classification of speech and gesture events as partial or complete and by checking the events timestamps.

Johnston & Bangalore [33] present a multimodal user interface for a corporate directory and messaging interactive systems. The system features two modalities: a pen-based and a speech-based one. The two recognizers (in charge of receiving the events produced by the input devices) send to the integration part (i.e. the fusion engine) a lattice representing the possible recognized strings and the possible recognized gestures. The fusion is described by means of a set of finite state automata representing a context-free grammar (one automaton for each modalities plus one for the fusion engine). [11]

The previously described systems differentiate themselves from the ones described in the Replication Phase, by allowing the concurrent use of two or more modalities and more importantly by making that use a synergistic one, according to the classification defined by Nigay & Coutaz [26] (see Table 2.2). Despite the advantages synergistic interactivity may bring, it also has its down-sides like increasing the complexity of the whole fusion engine, because it now has to deal with temporal constraints.

2.3.1.4 Theory and Automation Phases

Following the Empiricism Phase there's the Theory and Automation Phases. When the technology reaches this phase, hypotheses are formed about the causal systems underlying experience and developed as theories. In the automation phase, theories are accepted and used automatically to predict experiences and to generate design rules [11]. These two phases are bundled together because usually, in the field of fusion engines, each theoretical proposal is immediately followed by its integration in a system that takes care of the practical demonstration. One example of a system representative of these phases is HephaisTK [4] which will be explained in detail in section 2.3.2.2.

2.3.1.5 Maturity Phase

In Gaines's model, the theories developed when the Maturity Phase arrives have been considerably developed and assimilated and thus, are eligible to be used in a routinely manner, without any question asked. We can identify the moment a certain technology

reaches this phase, by looking at the deployment it has on the market, if it is being used in large practical application or in the field of safety critical systems.

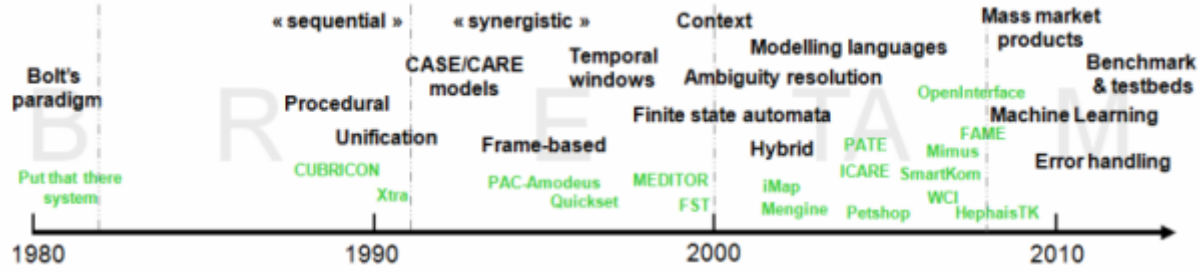


Figure 2.5: Evolution of multimodal systems and respective fusion engines according to the BRETAM model [11]

According to the vision of Lalanne et al.[11], multimodal interfaces have already reached the maturity phase. Indeed, the adoption of multimodal interaction is becoming the trend in entertainment systems mass distributed in the market, such as the Nintendo Wii [46] or Microsoft Kinect. Although the market is now ready for accepting multimodal systems, there's still work to be done on the field of fusion engines, their evaluation and quality of error handling [11]. Fig. 2.5 shows the evolution of multimodal systems / fusion engines through the decades and divided by the BRETAM phases.

2.3.2 Representational models

Evaluation of multimodal systems has mainly focused on user interaction and user experience evaluation [5]. Performing evaluations on these systems is of great importance to get insight about some given user interface, but when one is confronted with a multimodal interactive system, figuring out what to correct and how to correct it, can be a problematic issue. Benchmarking and evaluation of fusion engines is also a matter of great importance and will be further discussed in section 2.3.4 referring to the findings of Dumas et al. [5]. In order to explain some of the issues involving fusion engines it is essential to present two formal representations intended for modelling multimodal human machine interaction, the CASE model [25] and the CARE model [19].

The CASE model [25] focuses on the possibilities existent for combining modalities at the fusion engine level. It consists of four main properties: concurrent, alternate, synergistic and exclusive. Each one of those way expresses a different way to combine modalities in a multimodal interactive system. These properties will be explained with more detail in the next section.

The CARE model [19] approaches multimodal interfaces more from the user-machine interaction level. This model encompasses four properties:

- Complementarity - used by the user when multiple complementary modalities are

necessary to transmit the desired meaning to the system. Two or more modalities are considered complementary if all of them are necessary to transition the system to a certain state within a temporal window. A good example of the use of complementarity is in the ”Put that there” system [2] where speech and gestures had to be used together to represent valid action commands.

- Assignment - a modality is said assigned to a state change if it is the only modality capable of translating the intended meaning (e.g. the steering wheel of a car is the only way to direct the car).

- Redundancy - even if multiple modalities are to be used simultaneously, each one of them can be used individually to lead towards the desired meaning (e.g. user utters a “play” speech command and pushes a button labeled “play”, but only one “play” command would be taken into account).

- Equivalence - entails multiple modalities that can all lead to the desired meaning, but only one would be used at a time (e.g. speech or keyboard can be used to write a text).

2.3.3 Development Frameworks

Regarding the creation of multimodal interfaces, in the recent years, a number of tools have become available to fill the gap between the design & specification stage and the implementation process of a functional system [4]. More importantly, many of these developer toolkits enable a flexible integration of fusion engines. The next subsections will present two frameworks available today.

2.3.3.1 OpenInterface

OpenInterface [25] is an extensible software workbench for supporting the effective and dynamic prototyping of multimodal interactive systems. Lawson et al [25] distinguish the OpenInterface plataform from other toolkits intended for multimodal application design (e.g. ICON [39], ICARE [16], Exemplar [6]), because most of them either are limited to a specific technology or support a limited number of modalities. The approach used by OpenInterface is of an interaction, device and technology independent flexible solution for fast prototyping of multimodal systems through the facilitation and reuse of existing software and technologies.

2.3.3.2 HephaisTK

HephaisTK [5] is a toolkit which allows developers to quickly create and test multimodal interfaces. Its modular architecture (See Fig. 2.6) enables an easy configuration according

to developers needs (e.g. plugging in or out components such as recognizers). HephaisTK is designed to control various input recognizers, and more importantly user-machine dialog and fusion of modalities. A developer who wishes to make use of HephaisTK needs to provide two things: his application and a SMUIML script (Synchronized Multimodal User Interaction Markup Language). The SMUIML markup language [3] is an expressive, easy-to-read way of telling which modalities are used, the recognizers attached to each one of them, the user-machine dialog and the various triggers and actions associated to this dialog.

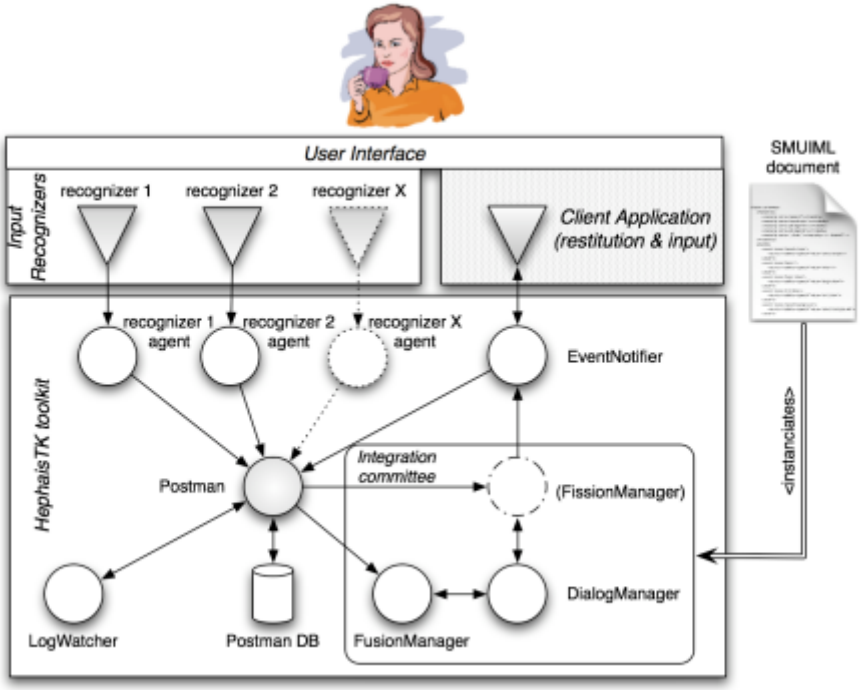


Figure 2.6: HephaisTK architecture [4]

The SMUIML document provided by application developers will serve the toolkit by defining: the messages traveling from the toolkit to the applications; the events originated by the recognizers that will have to be taken into account by the toolkit; a general description of the user-machine dialog. The communication between developer application and HephaisTK is done through a java class that applications must import, making use of Java listeners. The architecture modularity is assured by using JADE, an agent development framework. Each agent is responsible for the reception, annotation and delivery of the data provided by a recognizer. An agent responsible for a speech recognizer, would not only transmit information about what has been said by the user but also other possible relevant information (e.g. trustworthiness of the incoming data). After being received by the agents, the messages are then forwarded to an agent named Postman, which acts as "blackboard" of the system, where all messages get stored and are accessible to all the agents, acting like a publisher/subscriber system. Another great advantage of this central-

ized component is that it also manages timestamping of messages, thus dealing with all synchronization issues.

The modular software agent-based architecture of HephaisTK allows the toolkit to support various schemes of fusion, from rule-based to statistical to hybrid fusion schemes. The choice made by Dumas et al. [5] was to adopt a frame-based approach. The information integration works in an event-driven way: whenever the integration committee is alerted to a new event (in the case of fusion, incoming input), it confronts the knowledge received against the possible frames of knowledge of the current context. The FusionManager component always has notion of the application state thanks to its communication channel to the dialog manager (which follows the SMUIML script) and therefore always knows which frames to confront the new events with. A typical knowledge frame usually contains triggers and the respective actions for when they are activated. Moreover, frames activate following rules modeled from CARE properties [19], allowing temporal constraints to be specified.

2.3.4 Benchmark and Evaluation

As previously mentioned, errors in a multimodal system can originate from many different sources. Three important sources to consider are the modalities recognizers, the fusion engine and the user itself [5]. When problems such as ”The query did not produce the expected results” arise, it’s not easy to determine the cause of the error. For example, in a system using speech and gestures such an error could happen because of a poorly formed query, a recognizer issue, or delay in the system communication making the command not being properly fused. Giving users more complex and richer ways to interact with the system, will also affect the efficiency of user evaluation tests.

Dumas et al. [5] propose a “divide and conquer” approach to evaluate a multimodal interface. The idea is to adopt a step by step process, and base later evaluations on the results of the former ones. An example of such process would be to first evaluate the recognizers output, the performance of the fusion engine, fission engine and so on. When all of the “pieces” of the system have been tested against a sample of data they have to manage, the whole system can then be tested with real data provided by users. Since diverse toolkits for prototyping multimodal interfaces are now available, like OpenInterface or HephaistTK, which allow the plug-in of different fusion algorithms/schemes, it would be useful to define tests and metrics that could compare them.

In the next subsections it will be discussed testbed and performance metrics possibilities for the comparison of quality and efficiency between multimodal fusion engines, according to the vision of Dumas et al. [5].

2.3.4.1 Testbed for Fusion Engines

When dealing with modalities recognizers (from a developer perspective), one should take into account their error proneness. In order to bypass these errors, which take place before the fusion process can begin, the output of the recognizers can be manually generated and transmitted to the fusion engines. This solution is acceptable because the testbed main goal is to evaluate fusion engines and not recognizers. Furthermore, by controlling the input provided by recognizers it is possible to simulate incorrect outputs from them and analyse how fusion engines react to those anomalous situations. For all the output produced by the recognizers (a temporal and multimodal event stream), there will be an apropriate interpretation generated by the fusion engine. Then those interpretations will be compared to previously established "ground-truths" which represent the expected results. In this manner information can be obtained so that performance of the fusion engine can be evaluated (see Figure 2.7).

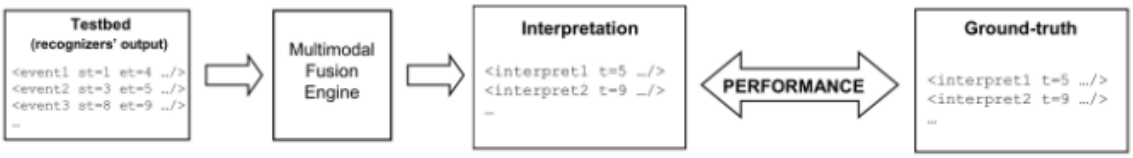


Figure 2.7: The event and interpretation flow in a multimodal fusion engine testbed allowing to rate its performance (adapted from Dumas et al. [5])

The choice made to represent the messages flowing through these components was EMMA (Extensible Multimodal Annotation markup language) [48]. EMMA is an XML-based markup language developed by W3C Multimodal Interaction Working Group, and is used for containing and annotating the interpretation of user input by multimodal recognizers. Examples of these interpretations would include translation of raw values into words, representing speech or gestures input.

The challenge of creating a set of use-cases focusing on major difficulties of fusion engines requires the identification and understanding of those same difficulties. To do so, Dumas et al [5] considered two formal representations for modelling interaction on multimodal interfaces, the CASE model and the CARE model (see section 2.1.5). Because the goal of the testbed was to measure how well fusion engines are able to interpret user's intention and how they make intentional use of the modalities at their disposal, the CARE model was chosen as the most suited for the task.

2.3.4.2 Metrics and Software Requirements

Dumas et al. [5] defined a series of quantitative and qualitative metrics to measure the overall quality of a multimodal fusion engine when confronted with a certain set of multimodal events. For each event it is considered three quantitative metrics [5] :

- Response time - time between the instant the fusion engine receives multimodal inputs and the instant it returns an interpretation of that input.
- Confidence - level of confidence of the machine response, based for example on confidence scores indicated in the EMMA testbed interpretation elements.
- Efficiency - success or failure of the fusion engine interpreting the testbed entries in a correct way. Efficiency is measured by comparing the interpretations produced with the ground-truth data.

Besides being able to respond quickly and efficiently to user requests, a fusion engine should have the following qualitative features:

- Adaptability - a fusion engine should adapt itself to context and user.
- Extensibility - a fusion engine should be capable of extending itself to support new or different input sources.

The testbed itself can also be characterized in order to help developers realize some of the limitations of their fusion engines. Some of these characteristics would be expressive power (developer/user-friendliness of the mechanisms used to configure the fusion engine), level of complexity (e.g. dealing with incorrect data from the recognizers vs straightforward equivalence of two input modalities) or problem type (keyword present in the testbed file, defining particular goals or features to be tested).

One of the most common techniques to debug issues in software development is logging. By checking system logs one can analyze user's behaviour and grasp useful information from it. Multimodal interactive systems revolve around time-sensitive architectures where all modalities should be properly time-stamped and synchronized. When several actions are performed sequentially it is advisable to establish temporal thresholds for time-stamping start and end of each input signal, so the real intent can be correctly interpreted. Another example of the criticality of time in multimodal systems is when two or more modalities are used in parallel. In those situations it is important to understand the correct order of events because different orders may convey total different meanings. Since all of this temporal data shows itself to be so important in analyzing what is happening in a multimodal system, it is safe to conclude that logging mechanisms are essential to the benchmarking of fusion engines. Dumas et al. [5] affirm that a system capable of logging input events and fusion-related events (timestamped), should be able to make use of the proposed testbed.

2.4 Summary

This chapter presented several aspects of multimodal interfaces as well as its underlying multimodal fusion engines. It has been seen that by providing multimodal interfaces with

adaptive features, they can support a larger spectrum of users, including the ones with special needs (e.g. elderly people).

Fusion engines are one of the important components in a general multimodal system, in charge of combining information coming from all sorts of input devices. The fusion process efficiency may be further boosted, by using auxiliary information such as user models (containing data about users, their capabilities/limitations, interaction patterns, history, among others) and context awareness. Several approaches to achieve multimodal fusion were presented throughout the chapter, along with some of the most common techniques/architectures used. After an overview of the historical background of fusion engines and their respective contributions to the field, some of the most well known frameworks to rapid prototyping of multimodal interfaces were discussed. Some of the frameworks available today allow an easy integration of fusion modules in order to test different fusion algorithms, which is enabling further advances in the evaluation and benchmarking of fusion engines.

Chapter 3

User Trials Application

This chapter presents the User Trials Application an application that was thoroughly used during the first year of the GUIDE project. It consists on a multimodal application capable of setting up scenarios for assessment of user abilities and limitations.

3.1 Introduction

The GUIDE project is heavily user-oriented, focusing on a very specific niche of the population, elderly and disabled users. Because of this, the approach adopted in the first year of development was to focus almost exclusively in planning and developing technological solutions to assess user needs and requirements. To this end a user trials methodology was followed, which consists of facing the user with prototypes not only to obtain information from the user about preferences but also to observe the user interaction with the system taking into account the difficulties found in this interaction. In order to achieve these objectives the UTA (User Trials Application), a multimodal application, was created and used in two pilot studies and has brought some conclusions about the user thoughts on preferences on modalities of interaction, visual, auditory, motor parameters, multimodality options, cognition and avatar usage. These conclusions served not only to understand user requirements and needs in general, but also how to improve the application and the evaluation protocol for the subsequent testing sessions. The results obtained from tests also gave some insight on ideas to implement the fusion module in GUIDE. These ideas will be discussed later in the chapter in the section related to the analysis.

Because of the level of interaction necessary for extracting relevant information about how the user interacts with the system, the testing could not consist of making use of low-fidelity prototypes and therefore the UTA was created. In developing the application, we adopted a "Wizard-of-Oz" approach (more on this technique in section 3.3.1), that enables the person running the test to gather more precise information on how the user feels about the interaction he goes through. The next sections will cover the UTA in more detail, explaining the overall goals, architecture and how the two pilot studies that made

use of it were set up, the tests performed, their analysis and conclusions.

3.2 Goals

When dealing with usability tests, it is important to grasp the most you can about what the users do and think. The UTA, that had to be developed in GUIDE, is an application that allows the user to interact with a series of highly-customizable screens and controllable by the individual running the tests. The means of interaction included were present in a significant diversity (e.g. speech, gestures) as well as interaction devices (e.g. Wiimote, tablet, remote control), which allows a broader possibility of choices by the user. In order to cope with the different disabilities the users may have, the application allows to customize all of the test scenarios, by a well-defined script, defining not only which elements should appear on screen (e.g. buttons, images, video) but also their properties (e.g. width, height, text, volume). The main purpose of these tests is to collect data about users and therefore a very critical feature of the application would be performing logging operations in order to store this information (e.g. selection times, tests performed), so it could later be analysed and discussed. As said before, the UTA makes use of the well known research technique "Wizard-of-Oz". Using this technique allows to mitigate or eliminate some of the drawbacks that the application could have, such as a poor performance by the speech recognizer. Making an individual the "Wizard" of the system also helps the information retrieval, because he can, for instance, ask what the user thinks about the test, their actions, preferences, cognitive skills among others. The "Wizard" is also capable of controlling the execution flow of the tests which can turn the whole test process more efficient.

3.3 Architecture

This section will explain in more detail the architecture of the User Trials Application, namely its components and the flow of messages that circulate inside the system. Before we began to consider how to design the UTA, an assessment of a Java custom framework called MMX [17], oriented for making applications ready for multimodal input, was made to understand if some of its features would be appropriate to use in the upcoming work. One of the features that seemed to be working well and efficiently was the communication protocols which was based on a publisher/subscriber system. This system is a message broker, a message-oriented middleware server that hosts messaging destinations (i.e., queues and topics) for the purposes of asynchronous communication. Due to the simplicity of message exchanging provided by this message broker, it was decided to choose Java as the main coding environment for the core of the UTA.

As we can see in figure 3.1, this system involves various components, some of them

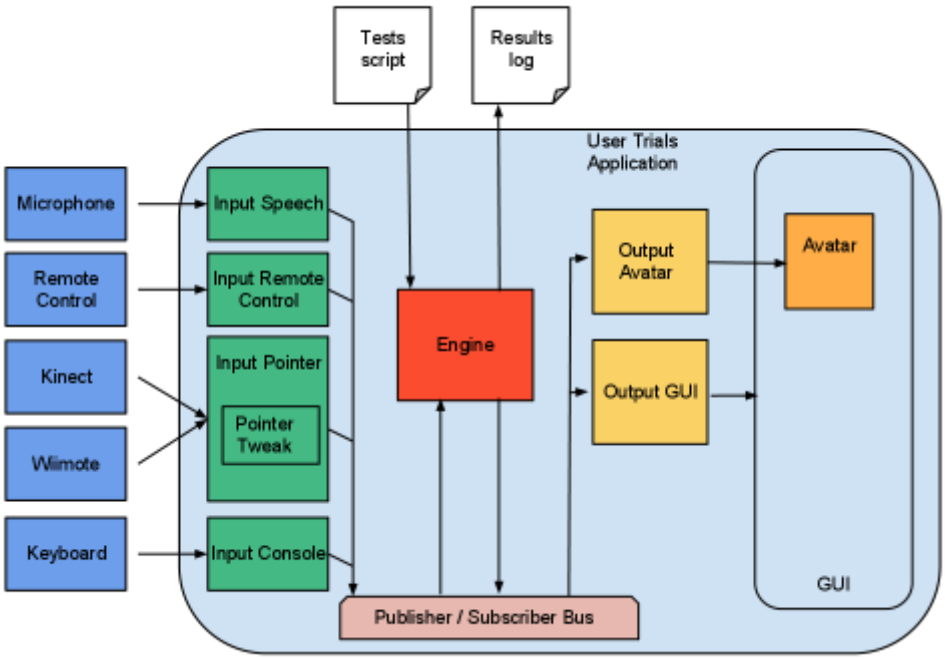


Figure 3.1: User Trials Application Architecture

inside and others out of the application itself. The process of interaction begins with the users, which has at his/her disposal a number of input devices, such as a remote control and microphone, which allow the transfer of input into the application. It is important to note in this diagram that the keyboard is not a device available to the user, but to the person running and controlling the testing session. The UTA was designed to handle both user and "wizard" input, because the latter is very important to empower the testing like for example ensuring that the proper tests are conducted for each user or making him repeat a certain task. For each kind of interaction or modality, the system has a correspondent adapter, which handles the raw input received from the devices and translates it to application logic objects. The "pointer tweak" module has the goal of adapting incoming pointing related input in order to enhance the quality of this modality, transforming coordinates that aid the user motor capabilities. More emphasis will be given to these enhancing capabilities in chapter 5, where components interacting with the GUIDE fusion module will be discussed. After the UTA receives the input it sends it to the "Engine" which acts as a Dialogue Manager, such as one described in the second chapter of this document, when a generic architecture of a multimodal system was described. This component controls the application state according to the commands received by the keyboard adapter, changing to a new state as requested. Another duty of the Engine is to parse the XML file containing the content and synchronization properties of each screen of the tests script. The structure of this script and how it is used by the developer will be exemplified in the next subsection. Every input detected by the sistem, sequence of states,

and other qualitative and quantitative data captured during the interaction between user and the UTA, are kept in an XML file that is constantly updated by the Engine, in an autonomous way or directly instructed by the developer.

When the Engine needs to make the application transition to a new state, it has to read the next test screen from the script file and communicate the adequate information to the output adapters. These include one avatar adapter and a "GUI" adapter. The former enables sending commands to an external avatar application, which was developed by another project partner, and that consists in a 3D virtual agent, or avatar, capable of performing animated gestures and speech synthesis, by receiving specific XML-based messages. The latter is the adapter that sends all the necessary data to render the final GUI to the user. This rendering is done by a C# component, which takes care of the output modalities of the application, placing visual elements on screen, adjusting their properties, and playing sound or other types of media.

The implementation of this system besides helping assessing the user requirements, was also useful in understanding how a multimodal system is implemented. As suggested by the generic architecture of a multimodal system (see chapter 2) the three main components are the multimodal fusion, fission and dialogue manager modules. In the UTA these were grouped in one central component, the Engine. Some of the tests included in the user trials involved the use of more than one modality simultaneously, which introduced the concept of fusion, how to handle multiple inputs that are to be combined in one result or decision. These ideas started to form the base for the later implementation of the GUIDE fusion core.

3.3.1 The Wizard-of-Oz approach

The Wizard-of-Oz technique is commonly used in computer science to perform an evaluation of unimplemented technology. This is done by using a human to simulate the response that is supposed to come from a computer. In most cases the "wizard" is invisible to the user, sitting in a back room, observing the user actions and simulating in real-time the responses from the system.

The UTA relies on the Wizard-of-Oz technique to some degree. The first big difference from the main concept behind the technique, is that the wizard does not only act as an "hidden" individual controlling the computer behind the scenes. On the contrary, the person running the tests must assume an active role in the whole testing session in order to either help the user with interaction or to register additional data that is not automatically kept in the results log. Using this technique has proved to be very useful in some tasks such as simulating speech recognition to allow total freedom to the user, letting him/her speak whichever words it would find appropriate in a certain context. The wizard besides being able to inquiry the user at any given moment, is also capable of controlling the application to some extent, including loading specific script files/tests, classify the user

performance in a given situation, simulate a user selection.

3.3.2 Tests Script

As said before, the tests planned for the user trials in GUIDE consisted on a series of interaction scenarios defined by a well defined script, which specified what kind of interface the user would be interacting with. This script was translated to a XML file according to the UTA logic and acted as the main source of information for the application, describing which elements to render on screen and how to do it. Figure 3.2 shows an excerpt of such a script.

```
xml version="1.0" encoding="utf-8" ?>
onfig width="1680" height="1050" filter="warping">
<visual>
  <question filter="" type="visual" number="welcome" screen="a" bgcolor="WHITE">
    <item type="label" synth="false" text="Visual Tests" fontsize="40" bgcolor="WHITE" fgcolor="BLACK" hlcolor="WHITE" x="0.006" y="0.300" width="1200" height="100" />
  </question>
  <question filter="" type="visual" number="Vis1" screen="a" bgcolor="WHITE">
      <item type="label" synth="false" text="Vis1 1/2" fontsize="20" bgcolor="WHITE" fgcolor="BLACK" hlcolor="WHITE" x="0.500" y="0.900" width="1200" height="100" />
      <item type="button" synth="false" text="CIENCIA" fontsize="40" bgcolor="WHITE" fgcolor="BLACK" hlcolor="GOLD" x="0.006" y="0.011" width="250" height="100" />
      <item type="button" synth="false" text="TV" fontsize="40" bgcolor="WHITE" fgcolor="BLACK" hlcolor="GOLD"  x="0.006" y="0.450" width="250" height="100" />
      <item type="button" synth="false" text="DEPORTE" fontsize="40" bgcolor="WHITE" fgcolor="BLACK" hlcolor="GOLD"  x="0.006" y="0.850" width="250" height="100" />
      <item type="button" synth="false" text="MUSICA" fontsize="40" bgcolor="WHITE" fgcolor="BLACK" hlcolor="GOLD"  x="0.800" y="0.011" width="250" height="100" />
      <item type="button" synth="false" text="VIAJE" fontsize="40" bgcolor="WHITE" fgcolor="BLACK" hlcolor="GOLD"  x="0.800" y="0.450" width="250" height="100" />
      <item type="button" synth="false" text="COCINA" fontsize="40" bgcolor="WHITE" fgcolor="BLACK" hlcolor="GOLD"  x="0.800" y="0.850" width="250" height="100" />
  </question>
...
```

Figure 3.2: Example of an UTA test script

The script is highly-configurable as suggested by the image above. It is possible for the wizard to group the tests into categories or modalities and inserting “questions” inside of them, in which each question corresponds to a GUI screen. The “question” can the be filled with item nodes that correspond to a certain element on a screen, including but not limited to, buttons, labels, video. Each one of these items will have its specific properties that can also be defined, such as text size, text color, background color, volume, use of sound synthesizer, width, height and position. By making use of this script approach, the wizard of the UTA can easily create an almost endless diversity of tests in order to capture the interaction experience of users and its relevant data.

3.4 User Trials

It is worthy to mention that prior to the user trials realization (involving the per-say practical interaction experience) a user survey was developed and conducted. This survey was an interview process which aim was to obtain data from each one of the participants to help develop a user model based on extensive user data and by dividing users in several clusters, each one with different characteristics. Since GUIDE is oriented for elderly users with mild to moderate impairments, these interviews also served the purpose of filtering out for the subsequent trials, users that possessed major impairments or no impairments at all. A group of questions was administered individually, by psychologists,

in face-to-face user interviews. Each interview contemplated more than 100 items combining objective and subjective assessment of the user capabilities, with information in the following areas: Socio-demographics; Attitudes towards technology; Coping styles with new technologies; Attributional styles towards technological devices; Daily living activities; Vision; Hearing; Mobility; Cognition; Personality traits. 46 elderly people (30 female and 16 male) with different disabilities were recruited. Their ages varied from 41 to 81. The average age was 70.5. The survey data was recorded by means of an audio recording and analysed after the survey.

The purpose of developing the UTA was to use it in the GUIDE User Trials, which constituted a primary source of user requirements, by observing elderly users in a controlled lab scenario, and make them interact with the application using several input devices. Qualitative and quantitative data was collected which allowed the identification of viable usage methods (e.g. gestures, command languages) of novel and traditional UI paradigms for the different impairments in the target groups.

The next sections will explain how the main study for the user trials was conducted, namely details of the subject group that took part in the tests, the interactive scenarios used for assessing user opinions and impairment-related characteristics comprising various modalities. The section dedicated to analysis will not state every result that was derived from the testing sessions, instead it will focus only in the ones relevant for the process of input fusion. The complete set of results and observations can be consulted in [18].

3.4.1 Set up

The participants in these trials included seventeen elderly individuals, aged from 55 to 84 (with the average being 65.8 years old). Regarding the gender, four participants were male and thirteen female. All of them were recruited in Spain, where the testing sessions took place, more specifically in Ingema facilities. Before starting to interact with the UTA, users were instructed on how to use the different available input devices so they could use them with relative ease. These devices included the Kinect, Wiimote and a regular TV remote control. This initial test also included a few questions related to preference of interaction and a small experimentations with a few screens so the user could try to select some items and start to get used to the devices he would have to interact with. The room set up was made so the user was sitting before a screen and speakers so he could perceive the output of the application (see Fig. 3.3) The wizard also possessed a laptop which he used to control the application flow and all of the variables already mentioned in the previous sections.

Every assessment was recorded with two cameras, one of them focused on the user and the other on the TV screen. Analysis of the user-trials was made from analysis of these videos. The measurements taken in the main study were: number of errors; time

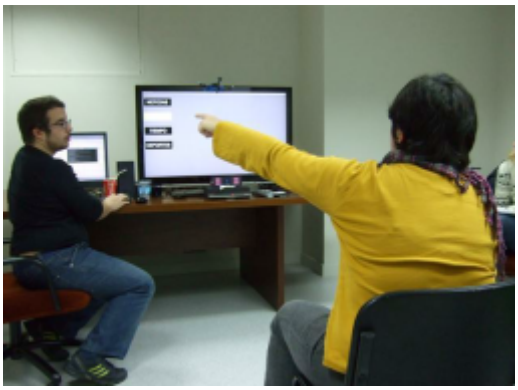


Figure 3.3: User Trials Set up

Figure 3.4: An User Trials test subject

rate (e.g. time it would take a user to select a certain button); observation of participants actions and behaviour. Before the start of the trials, the application along with a nearly finalized test script was tested in our facilities in Lisbon. This step was important to assure that software or hardware issues would not arise during the trials with the users. All the required material was then sent to Ingema so their team could familiarize with everything and be able to run the trials.

3.4.2 Tests

Defining different types of tests for the UTA is not an hard task, due to the high customization allowed by the supported XML format. In each user-trial every user was presented with all possibilities of interaction (pointing with the hand, Wiimote, voice or TV remote control) and asked to perform a series of tasks related with TV interaction. The tasks were divided into several scripts concerning different types of interaction or different UI elements. The next subsections will address each of the modalities present in the scripts and the general ideas behind the tests performed around each one of them.

3.4.2.1 Visual

For the tests concerning visual features, the user was confronted with many scenarios. They were presented with screens containing buttons and were encouraged to tell how

they felt on the positioning and inter-spacing. They were also request to comment on their color preferences (either font and background). Another visual property that was tested was font size of texts and the color contrast between them and the background (see Fig. 3.5). For each one of these tasks the test subject could perform selection commands by using whichever modality or interaction technique/device he/she found most comfortable and enjoyable, that being for example, voice, pointing, button presses or a combination of those.

Figure 3.5: User Trials Set up

3.4.2.2 Audio

Audio tests consisted on playing a series of different sound clips with different volume that were progressively adjusted. They were asked to repeat after the sounds to assure that they heard what was spoken by the application. After this, the user would be questioned about volume preferences, stating which level of volume he/she thought it was more adequate. These tests were repeated with TV sound playing simultaneously on the background, as well as with different voice genders (male and female).

3.4.2.3 Motor

To understand which gestures are preferred by the user, they were asked to perform a series of one and two-handed gestures both in front of the TV and on a tablet PC to assess which ones were most comfortable for them. The other set of motor tests consisted on confronting with the option of using a remote control (either the Wiimote or a regular remote control) or pointing gestures to select items on a menu, taking into account button placement. Finally the use of pointing motion was performed with and without tweaking algorithms (e.g. gravity wells) to understand which option provided a more accurate motion of the screen cursor considering motor impairments such as trembling, that could affect accuracy or speed.

3.4.2.4 Cognitive

Cognitive tests were used in the form of “games” in order to evaluate the participants visual memory and attention capacity. The game consisted on showing the user a screen

where a series of images would appear for a determined number of seconds. After that the images would disappear and the user would have to indicate where a specific image was located. The tests differed among them by customizing the number of images on screen at a given moment, the time interval during which they were visible and how much time the user would have to provide an answer.

3.4.2.5 Avatar

Different screens were presented in such a way that on one screen a audio message was transmitted by an audio file and then in next screen an animated avatar would appear and repeat the previous message through its speech synthesizer. Other aspects of the avatar were also discussed with the users, namely the types of shot of the avatar that they preferred (i.e. close-up, half-avatar, full-avatar body).

3.4.3 Results and Analysis

The following items represent the conclusions formed from the user trials and that are relevant for the fusion process in the GUIDE core:

- Filtering tweaks: Users that experimented pointing interaction through the Wiimote or by finger-pointing concurred that by using the gravity wells filter they were able to make selections faster and more accurately. This filter, as the name suggests, creates "gravity wells" around interactive elements on screen, attracting the cursor to them. The closer a cursor is to a certain element, the stronger it gets pushed towards it. For instance, if a cursor position is near the vicinity of a button, then the pointer will automatically move towards its center. Using this technique, users were able to select an item in an average time of 3515 milliseconds, but without the use of the filter this duration would be of 5120 milliseconds average. Even though the fusion module is not responsible for the algorithms that rule this pointer tweaking techniques or the component itself, it is important to note that using these in the pointing interaction improved the overall performance of the user and therefore of the system. When dealing with inputs, the fusion module is very dependent on synchronization issues. Receiving a certain command after a determined threshold may seriously impact the result produced by the fusion core. So, if the effectiveness of the user can, in any way, be improved either by speed or accuracy, it is also beneficial for the fusion process.

- Multimodal interaction: More than half of the users (53%) showed desire to have both multimodal input and output when they are using the system (even if only in specific contexts of interaction). Moreover, 59% of users said they want to interact using more than one modality at the same time, leaving only 29% wanting to

interact using one single modality. Additionally, 82% of users said they preferred multimodal feedback from the system, and only 18% say they wanted information present in only one way (just visual information, or just audio information). This data clearly suggests that user do want to interact in a multimodal way, combining a series of inputs when interacting with applications. The implication of this to the GUIDE fusion core is arising a need for preparation of multiple inputs at any given time. The idea is to create many ”scenarios” inside the fusion core that allow for the maximum number of modalities combinations in a certain application context. This will be more clear in the fifth chapter when the architecture of the core will be explained.

Another interesting fact that could be observed in the user trials was that the users sometimes had the tendency of using certain voice commands intuitively, such as words like “select”, “yes”, “no”, “this”, and “confirm”. This suggests that the fusion core should not only be aware of the commands available for a given time for a given application context, but also keep in mind that some special cases may have to be continuously available to be “triggered”, so that when the user provides input that is not directly related to the current context but to the set of pre-defined “GUIDE commands” the system reacts accordingly. This forms an application-independent awareness on incoming input that the fusion core must also have in consideration.

3.5 Conclusions

This chapter described the User Initialization Application, a multimodal application capable of supporting a set of input devices and through a customizable XML script render different forms of output to the user. The application makes use of a “Wizard-of-Oz” approach to extract the most information possible (e.g. what the user thinks about a certain task) and tackle issues related to technology such as poor performance by modality recognizers.

Developing the UTA was a primordial task during the first year of the GUIDE project because of its focus on user requirements gathering. This process included, besides a series of user interviews, user trial sessions where elderly users interacted with the UTA, accompanied by a “wizard” that supervised and ran the tests.

UTA’s architecture encompasses many components, which include adapters to receive the raw input from the recognizers and transform it and forward the information to the “Engine”. This component acts as the dialogue manager of the systems, keeping track of the current state of the application and initiating state changes when needed. Being a central component of the system, the “Engine” also performs other functions such as parsing and interpreting the representation of the current state, store results about tasks, and send the appropriate instructions to the output adapters which will deliver the commands to

their respective output renderers.

Defining which tests a user should take is easy because of the customization allowed in the configuration files. By adding specifics tags to the file, the "wizard" is able to render media elements such as buttons, labels, videos, audio clips or animated responses from a virtual avatar.

Being GUIDE a multimodal system it was important to include in the user trials very distinct types of tests, that targeted different modalities, either used in an unimodal fashion or by combining multiple modalities. These tests types included vision (e.g. preferences on button spacing, font size and color), audio (e.g. preferences on level of volume), motor (e.g. performing a series of gestures and pointing motions to understand which were the least burdensome, testing of cursor motion algorithms), cognitive (e.g. testing memory spatial skills and reaction time).

The UTA is an application that is not limited to only the scope of the user trials tests, in fact it can be extended to represent some other kind of multimodal applications, but it was essentially developed to be used in the GUIDE User Trials. The results of the survey and the user trials confirmed that elderly users can indeed have particular nuances regarding their disabilities and limitations. They have demonstrated certain preferences over some modalities. This is something that reinforces the need for a project such as GUIDE and why concepts such as multimodal fusion are worthy to study and discuss.

Chapter 4

User Initialization Application

The UIA (User Initialization Application) is a multimodal application that will be part of the GUIDE framework and that has resulted from the need of sharing knowledge between user and system. GUIDE needs to know the user's abilities in order to adapt the interaction to its users. The user needs to know what GUIDE allows him or her to do interaction wise.

4.1 Introduction

The previous chapter showed that elderly users can be very distinct in the type of limitation and disabilities they present. The survey realized prior to the user trials encompassed many question addressing multiple variables but all of that data was analysed for highly correlated variables and then a small number of general variables was found and used to represent vision, hearing, motor and cognition impairments. This resulted in the creation of three types of user profile clusters that translate into low, medium and high impairments profiles.

Since GUIDE is heavily user-oriented, associating a user with the most adequate profile is a task of the most utter importance. If the framework wants to efficiently adapt to its users then it must possess the most accurate information possible about the individual that is interacting with the system. The UIA is a multimodal application that was developed to fulfil this objective. This application is the first contact between user and system in GUIDE. The user is presented with several screens, aiming to evaluate how well he performs in tasks that are related with a specific modality (e.g. vision, hearing, movement). Theses tasks are tests similar to the ones implemented in the user trials, targeting, besides users capabilities, their preferences on certain aspects of the applications such as font size, font color, volume or button spacing. The ideal scenario is to extract all the necessary information and make the user feel like he is not taking some kind of test. When this process is over, the UIA assigns the user to a certain user profile cluster, which will serve as a reference for future adaptive processing that any component of the framework may have to do. Even though the goal of the UIA is centred in just assigning users to profile clusters,

this profile is likely to suffer changes over time because it is expected that the user evolves in some capabilities as he/she uses the system more and more or other factors affect their performance regarding some modality. Another very important purpose of the UIA is to familiarize the user with all the devices and interaction possibilities he has when using any TV web-based application inside the GUIDE set-top box. Besides all the information the UIA can explicitly give to user about interaction mechanisms available through output channels, he also implicitly learns by using the input devices right away in the UIA. If we think about elderly users, we can say that a large part of these type of the population is not yet used to interaction with TV web-based applications, and so by providing this initial insight about the system possibilities and how to operate with it, may result in a wider acceptance and overall efficiency. The adaptive features of the GUIDE framework start in the UIA, because as the user is selecting the preferences, the application gradually adapts itself and changes are visible throughout the profile assessment phase.

The application that was developed this past year will serve as a prototype for the final User Initialization Application that will be running within the GUIDE framework, even though it works as a good representative of the ideas discussed above. This chapter will focus on the prototype created, discussing the architectural approach and the implemented tests.

4.2 Architecture

This section will explain the architecture of the UIA (see Fig. 4.1), its functioning and the components that interact with it. As mentioned in the previous section, the UIA is a web-based application, which at the moment are the target application types for the developers that will be using the GUIDE toolbox. For this reason and because it is the first contact of the user with the system, the UIA provides a good example on how the final applications can work inside the GUIDE framework.

The interaction devices that can be available to the final version of the UIA are the same as for any other application making use of the GUIDE framework. In our prototype however, we choose to only use the Microsoft Kinect for pointing movement recognition and an open-source speech recognizer. The reason for this was that the tests that were planned to be implement did not require extra ways of input. The final version of the application may however, make use of all the input possibilities to gather the most complete set of information from the user.

The version of the GUIDE core implemented kept most of the components utilized in the UTA and with similar behaviours. A Dialogue Manager plays the same role as the "Engine" did in the UTA, keeping track of the current state of the application and selecting the next state it should transition into. As said before, the UIA keeps adapting itself during the course of the interaction, reflecting the preferences of the user into the output

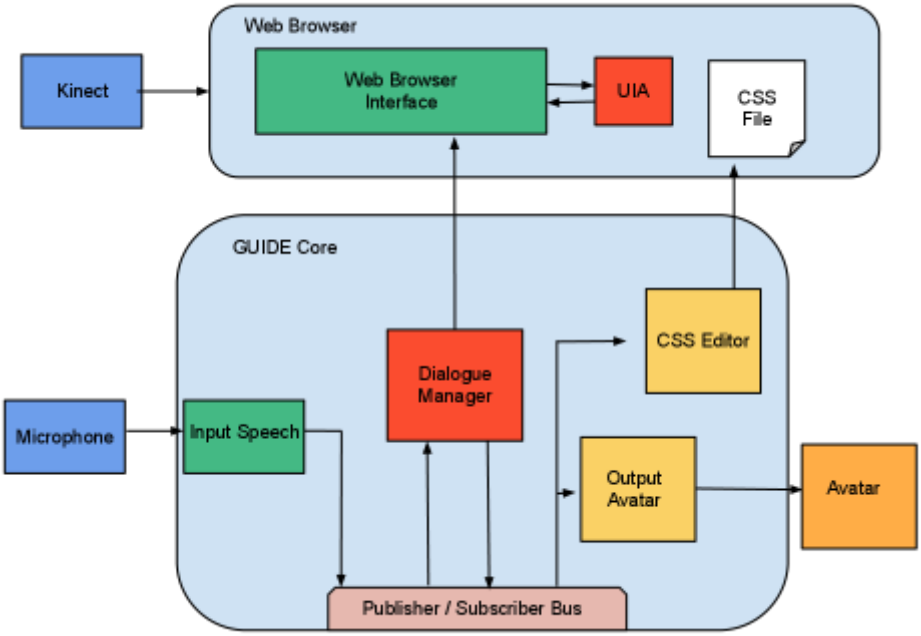


Figure 4.1: User Initialization Application Architecture

modules. For this to happen the Dialogue Manager keeps this information adapting its behaviour and communicates visual changes to the "CSS Editor", a component responsible for making the proper changes in the CSS file that is read by the Web browser and displays the Web pages to the user. The "Web Browser Interface" is an applet, used for performing the communication between the UIA and the "GUIDE Core". Even though the supported modalities of this prototype only include pointing and speech commands, multimodal fusion is still done by combining events from these two modalities. An example of such events is a voice command and a pointing event that are fused together to perform a selection of a button or image. The user is not obliged to combine the modalities available but he's able to and thus the system can already start to assess the user preferences on interaction possibilities.

4.3 Tests

When the users are making the first contact with the GUIDE set-top-box, they must go through the UIA to later use other applications and have them adapt to their respective user profiles. The UIA first presents some introductory screens to give a context to the situation and to enlighten the user on how he can interact with the system, by explaining the modalities and devices available.

Then some initial tests consisting of simple selection tasks follow, and serve as a tutorial, that allow the user to have a sense on how to use the available devices (See

Fig. 4.2 and Fig. 4.3). It is important to notice that all of this interactive experience encompasses not only multiple input modalities but also output modalities (i.e visual and audio). The animated avatar acts as a guide for the tutorial instructing the user with speech commands.

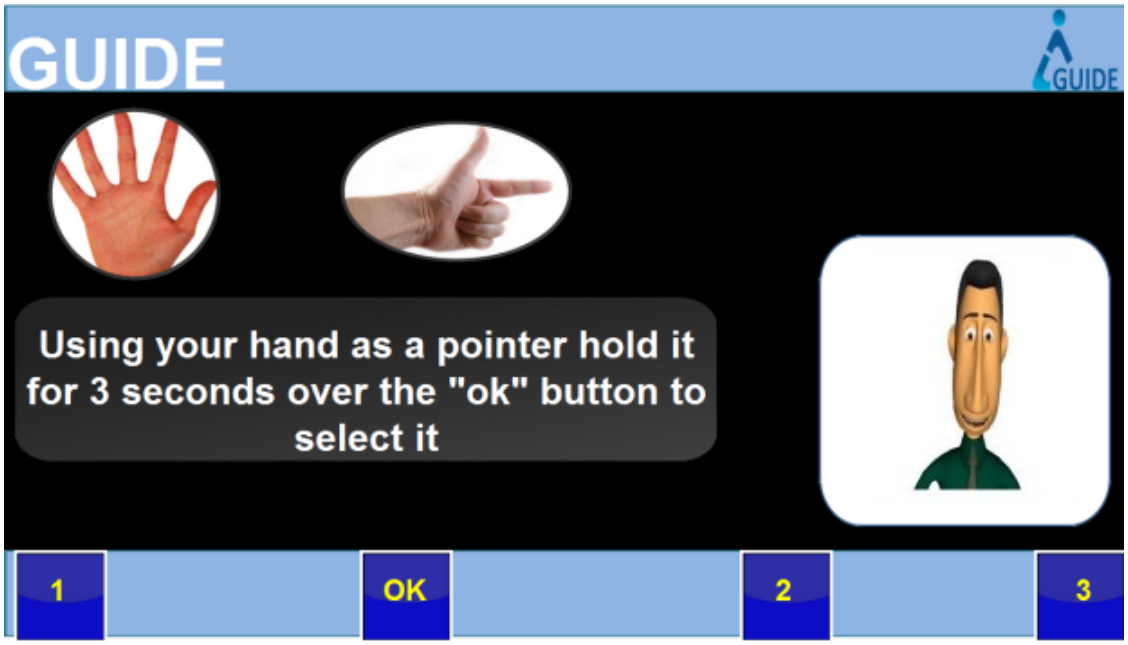


Figure 4.2: UIA pointing training

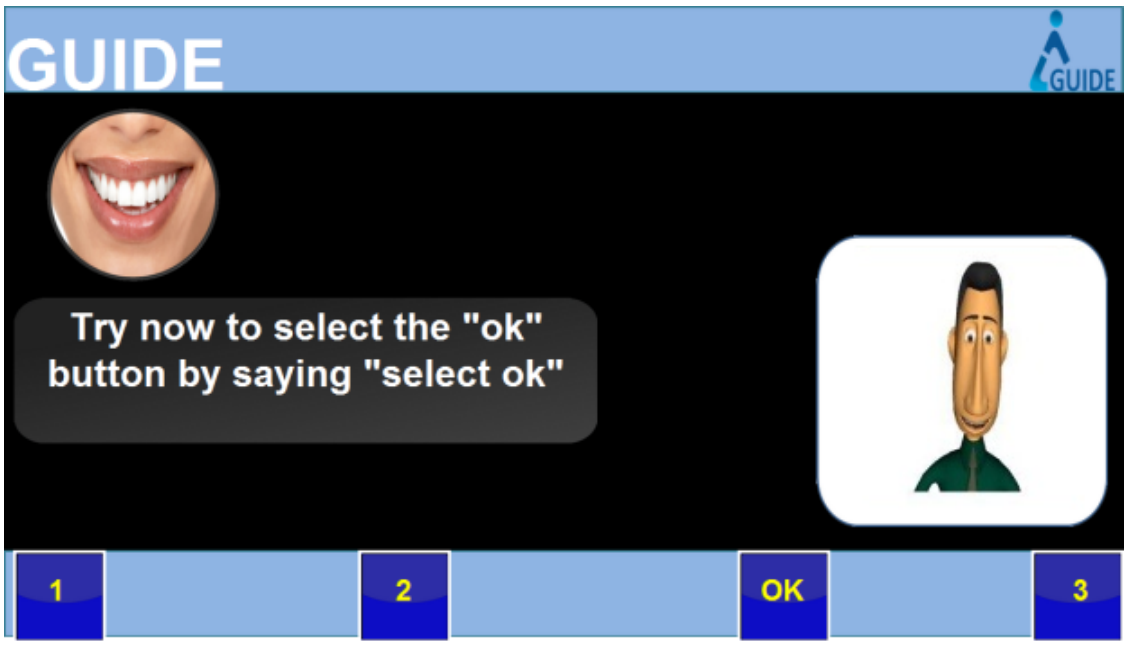


Figure 4.3: UIA speech training

The next subsections will briefly discuss the tests that were implemented in the UIA. Many of these tests are very similar to the ones used in the user trials conducted before

the UIA development. These can exemplify the kind of tests that can be included in the final version of the applications since it covers the most relevant modalities inserted into GUIDE context.

4.3.1 Visual

Visual output is perhaps the modality most used in most unimodal systems to present information to the user and so testing this modality is quite important. First the user was prompted to decrease the font size of a text until it felt no longer comfortable to read it. This task intended to find the optimal font size for the reader (see Fig. 4.4). Other visual tests included changing variables related to text and buttons, namely text font color, vertical and horizontal inter-space between buttons and other buttons-related properties like text and background color.

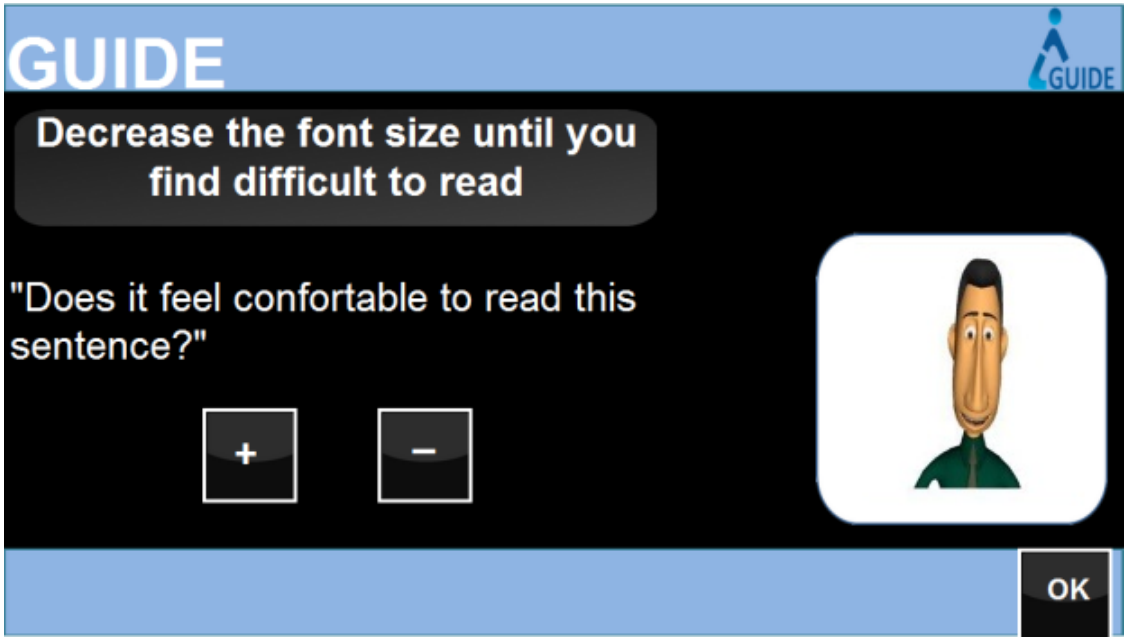


Figure 4.4: Font size selection

It is important to note that as the user chooses these visual preferences the subsequent tests already reflect this changes in the user interface, making the UIA self-adapt during the process of interaction increasing the efficiency of the data retrieval and comfort of the user.

4.3.2 Audio

Testing the audio capabilities of the user in this prototype is done in the form of two tests: in the first one the avatar instructs the user to press a button so a phrase is spoken and to continue to do so until it is no longer possible to hear it. The main goal of this task

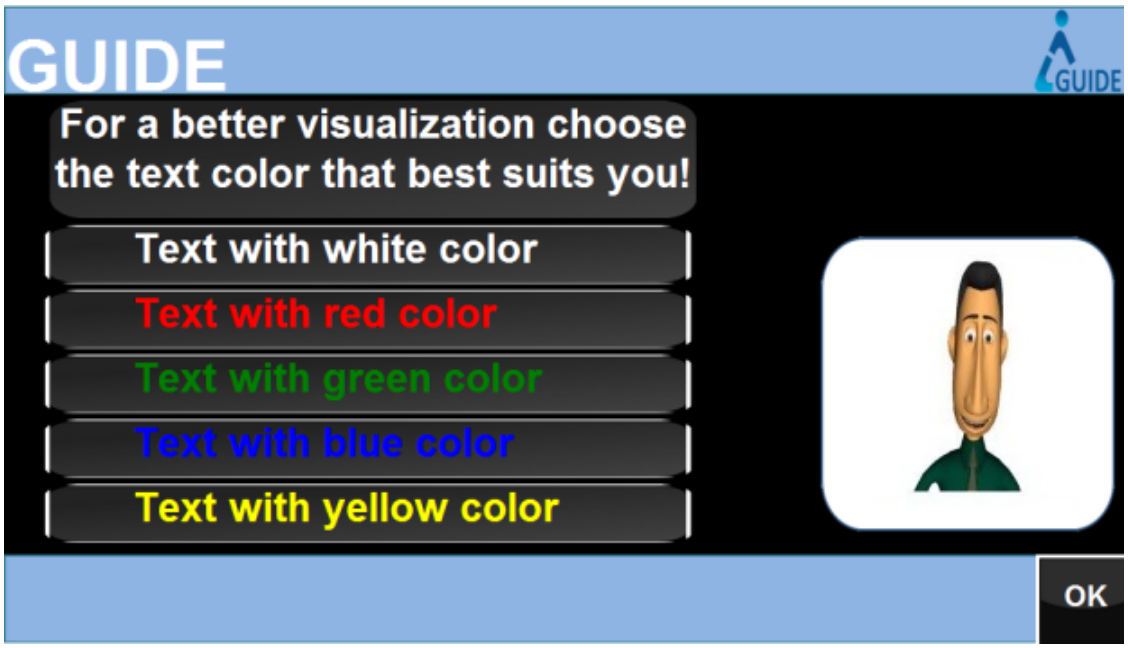


Figure 4.5: Font color selection

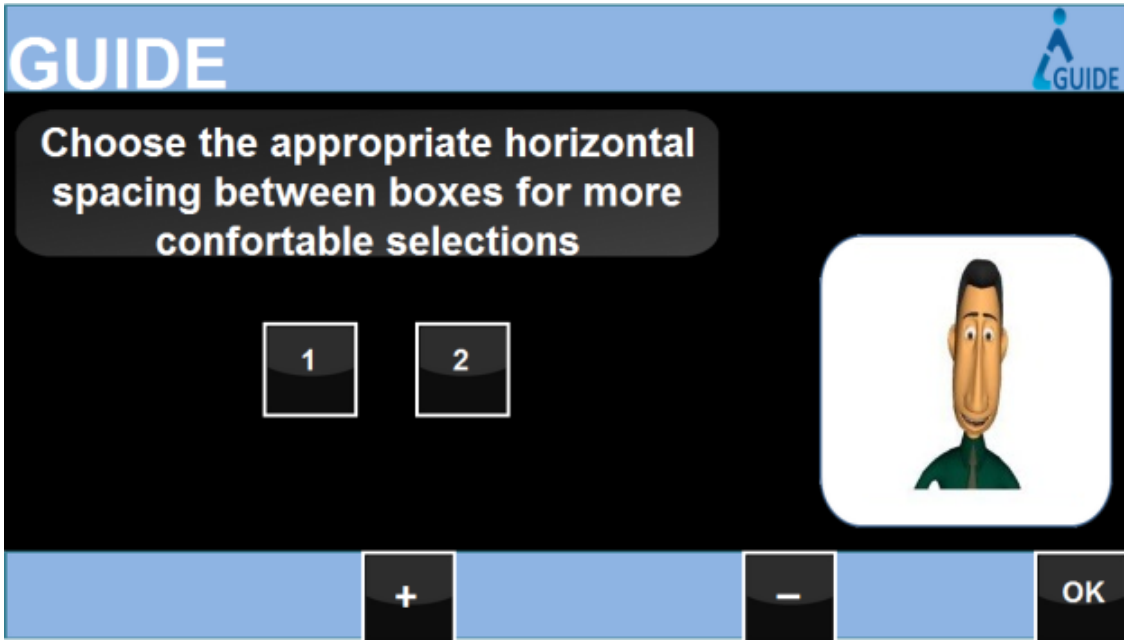


Figure 4.6: Button inter-spacing selection

is to determine the minimum level of volume allowed for that specific user. The second test is similar to the first, but it adds background noise to the synthesized phrase to test the contrast of volumes originating from different sources. This is done to simulate an environment where, for example, the user would be watching TV and then an application-related audio event would occur on top of the previous audio (or vice-versa).

Figure 4.7: Testing audio levels and contrast

4.3.3 Motor

The motor tests consisted on instructing the user to perform gestures using their arms, stretching to far edges of the screen, holding that position for a certain amount of time. Then the user would be inquired about the actions just made to assess their difficulty and result to the user current physical condition.

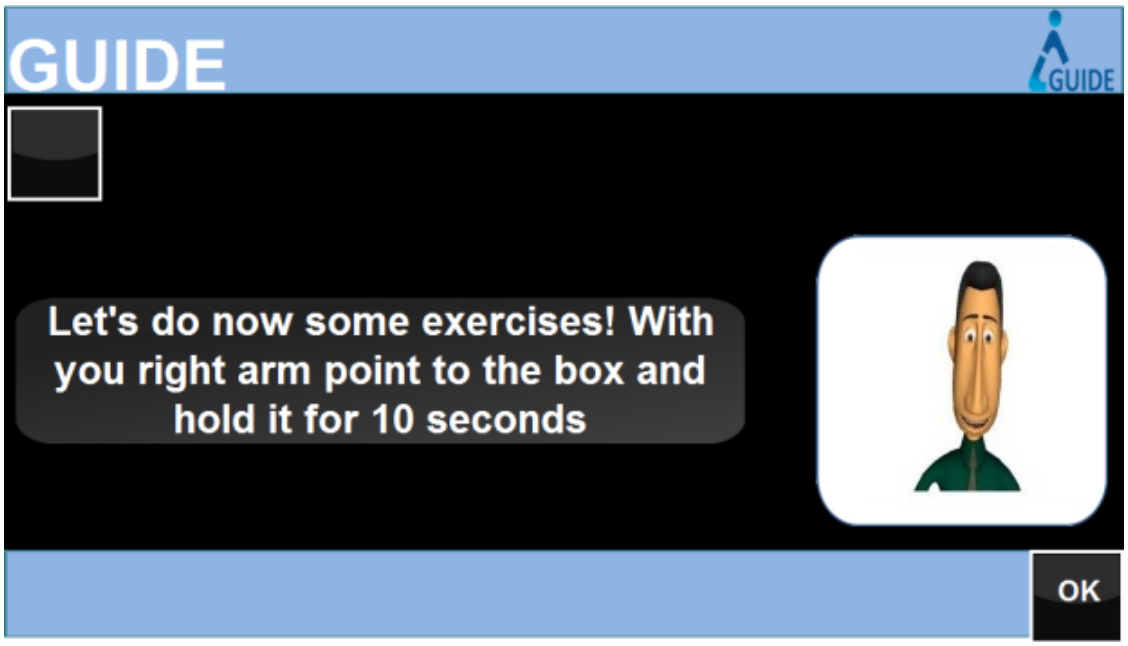


Figure 4.8: Testing motor limitations

4.3.4 Cognitive

To represent a possible cognitive test for the UIA, a sort of memory game was implemented, in which the user see in the screen a number of images that are occluded after a certain time and then the user has to guess the location of a randomly chosen picture.

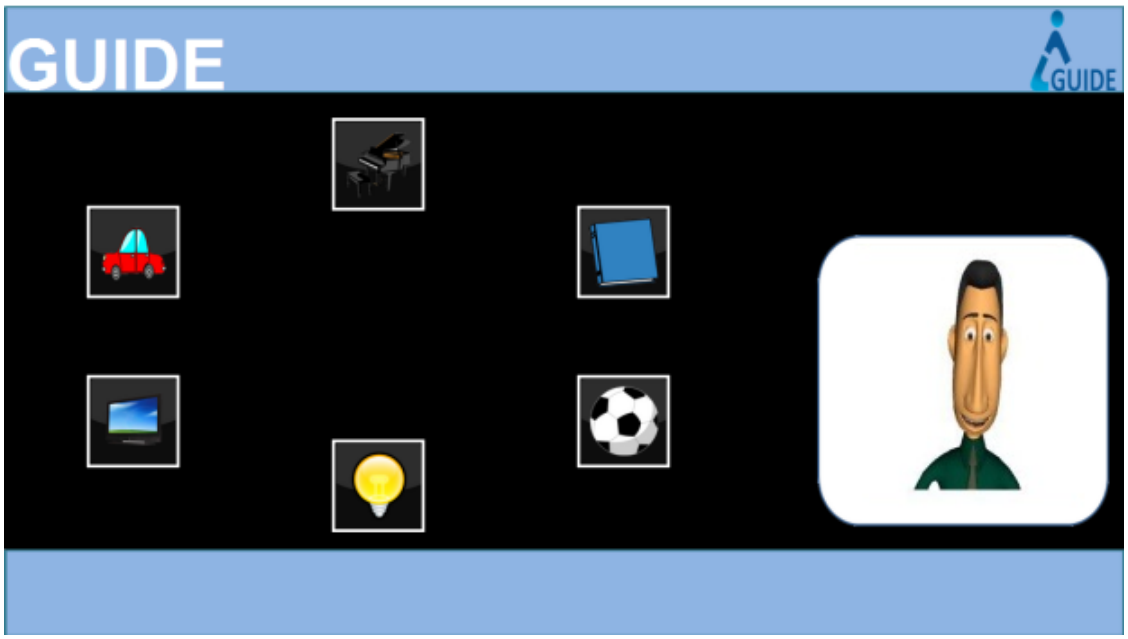


Figure 4.9: Cognitive test: Before images occlusion

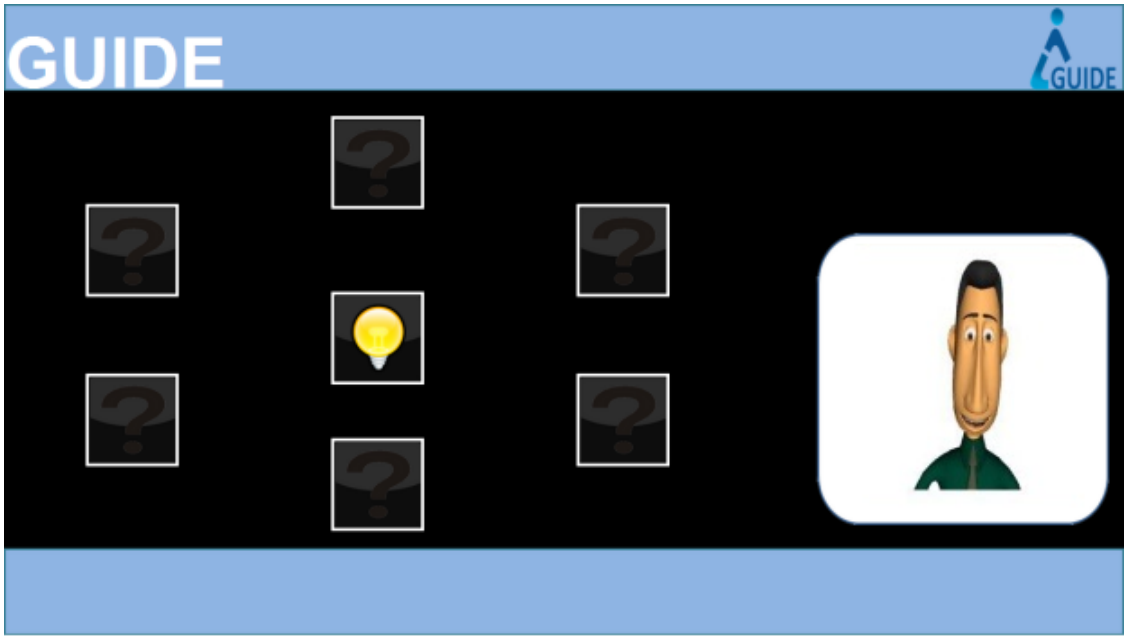


Figure 4.10: Cognitive test: After images occlusion

4.4 Conclusions

This chapter introduced the User Initialization Application, a multimodal application that will be an integrate part of the GUIDE framework, with the purpose of extracting user characteristics (including disabilities, limitations and preferences) so a user profile is assigned to each user before they utilize the system. Doing so, will aid many of the framework components performing adaptation, changing their behaviours to best fit the user needs.

A goal that was proposed by the project consortium during this past year was to have a prototype of the UIA functioning so the concepts behind it could be demonstrated. This objective was undertaken by FCUL and we proceeded to develop the requested prototype.

The architecture implemented on the UIA prototype included interaction devices such as the Microsoft Kinect for pointing motion recognition and a microphone for speech commands. The Dialogue Manager of the system receiving input events from these recognizers would then activate the proper response by the output modules. Being an application that is supposed to run in the GUIDE core, the UIA is a web-based HTML/JavaScript application that is also target of adaptation. The representation of the UI is formed by a CSS file that is frequently updated by the Dialogue Manager during the application execution in order to reflect user preferences. As the user progresses through the tests included in the application, which address vision, speech, motor and cognitive limitations or impairments, the applications adapts itself by turning the font bigger, raising up the volume, adjust buttons distance between each other, among others.

Chapter 5

GUIDE Fusion Core

In this chapter, the GUIDE fusion core is introduced in more detail. As a system grows in terms of interaction mechanisms available, so does the amount and variation of information received. For this reason, a way of correctly interpreting all of this data is needed, along with adaptation mechanisms that make the interaction experience the most adequate for each user.

5.1 Introduction

The GFC (GUIDE Fusion Core) is one of the main components in the GUIDE framework. Like a typical fusion engine (see chapter two) the objective of the GFC is to receive the incoming input from the user, which may be expressed by means of different modalities, combine the information received and forward the appropriate response to the next component in charge. One of the features that strongly characterizes the GFC is the drive and necessity for adaptation to both user and context, as it happens with most of the other components in the framework. The third chapter of this document showed the results obtained from the user trials that took place in the first year of the project. These results, together with the correspondent analysis, confirmed that users opt to interact with an application in a multimodal way when they have the chance, therefore the existence of a fusion engine in GUIDE is considerable important. The existence of an adaptive behaviour by the GFC can also be justified with the results produced from the survey that took place before the trials. These encompassed the categorization of users into three clusters with distinct variables. These clusters represented initial user profiles that contain information about users, namely their impairments and limitations, data that is crucial for the task of adaptation. Since the users can be differentiated by so many variables, the need for adaptation can be considered crucial for a system like GUIDE, that aims to tackle the accessibility issues of elderly users. The User Initialization application, discussed in the fourth chapter, focused on showing how all of this information about users can be collected and stored as user models to be later accessed by other components of the system to perform adaptation-

related tasks. So, as far as the fusion module is concerned, there are three main sources of information that are relevant for combining input and adapting to user: the information concerning the current context of an application, allowing the fusion engine to know how the user is supposed to interact (e.g. available commands, interactive elements currently on screen); the input received by user using one or more modalities; the user model that is assigned to each user by the UIA.

The two main sections of this chapter will be about the on-going implementation and evaluation of the adaptive fusion module of GUIDE.

5.2 Implementation

5.2.1 Architecture

In any system that supports multiple modalities for input, such as gestures or speech, there is a need for a multimodal fusion core. This component of the system is responsible for receiving the incoming input from different sources, combine that information according to specific context or user sensitive-information and making an interpretation out of it.

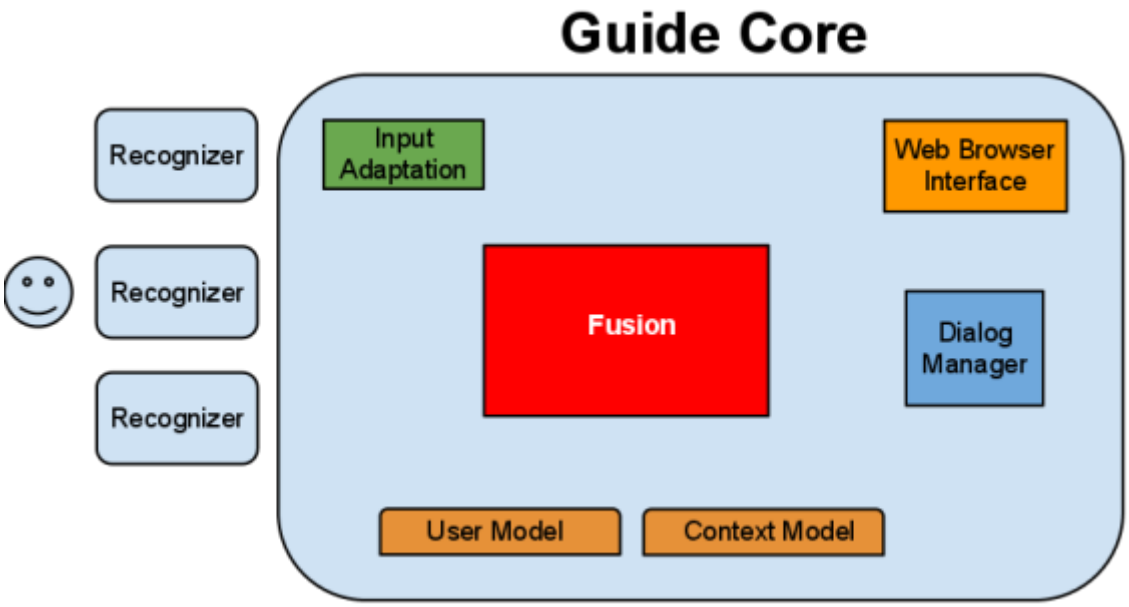


Figure 5.1: The GUIDE Fusion Core Architecture

As mentioned in chapter two, there are different approaches when there is a need to implement fusion in a multimodal system. The main options available are data-level, feature-level, decision-level and opinion-level fusion [8]. The latter two, are specific for cases where loosely-coupled modalities have to be employed, and so they are the only relevant possibilities considered for the GUIDE project.

The diagram of Fig.5.1 shows how the fusion core relates itself to other components in the GUIDE framework. Note that this image does not reflect the complete architecture of the toolbox, only the components that have an impact on the workings of the fusion core. As we can see in the aforementioned diagram, the fusion core, is a complex component of the GUIDE core, which has to communicate with several modules in order to work and have a purpose. These can be divided in two groups, the ones inside the framework and those outside.

5.2.1.1 Components outside the framework

- Recognizers
  First of all, and perhaps most importantly, outside of the guide core lie the recognizers, pieces of software responsible for capturing specific forms of input (e.g. having a recognizer for gestures, pointing or audio input) and forwarding that information to the fusion module. As said before, there are different types of fusion available, and therefore the information sent by these recognizers can also vary in quantity, quality and format, depending on the choice the developers make. Since only decision and opinion-level make sense in the scope of GUIDE, the recognizers will have to produce "hard-decisions" or "opinions". The main target audience of the applications which will use the GUIDE framework are elderly users. Sometimes these users have trouble transmitting their real intent and for that reason is expected from the system to able to deal with unwanted input or information that does not make much sense. For this reason, the capability of representing uncertainty is very important and opinion-level fusion is based on this concept. Instead of the recognizers providing hard-decisions, they deliver opinions with a certain confidence-level score, which will allow the fusion module to assess which decisions are more reliable and should be taken into consideration.

5.2.1.2 Components inside the framework

- Input Adaptation
  The "Input Adaptation" component, residing inside the GUIDE core, works pretty much as a recognizer, from the fusion point of view, because it essentially forwards input from the user to the GFC, with slight modifications. This component has the task of receiving the input related to spatial coordinates and treat this information. The main idea behind it is to use certain algorithms (e.g. gravity wells, cursor movement smoothing) in cursor movement to enhance the quality of interaction that the user has with pointing actions. Some of these algorithms have shown to improve users accuracy in selection tasks (see chapter three). The gravity wells algorithm helps the user select interactive elements by attracting the cursor to their center, if its location is near their borders. Other algorithms provide aid in other

ways such as countering blunt movements by the user, slowing down the cursor speed by creating a "counter-acting force". The relation that the fusion module has with the Input Adaptation is similar to the one with the external recognizers. The only difference is that, if the framework decides so, pointing events will pass first through this component instead of being directly sent to the fusion. Therefore the GFC is able to receive two forms of pointing input values, raw and processed or adapted values. By applying these algorithms that enhance pointing interaction, the "Input Adaptation" components is already performing adaptation to user.

- Dialogue Manager
  The Dialogue Manager is one of the central components of the GUIDE architecture (along with fusion and fission) and is responsible for controlling the state of the application and managing the communication between components. One of the main tasks of the dialogue manager, concerning fusion, is to receive from the WBI (Web Browser Interface), a user interface representation of the current state of the application, augment the information within, and send it to the fusion module. Once the fusion has this information, it gains knowledge about all the interactive elements that are currently on screen as well as their relevant properties (e.g. width, height, position). Knowing each element which is possible to interact with, in the current context, the fusion core can then make preparations to correctly understand the input given by the user.

- User Model
  Any user that interacts with the GUIDE system will have a specific profile, which is assigned to him early on, when he has the first contact with the user initialization application. This model holds information about user disabilities, accessibility issues and preferences. When the fusion receives the current state representation (the UI elements and properties) from the dialogue manager, it can make queries to the user model so a better understanding of the user can be made, and interpretations can be formed based on specific user-information. Since this model can be updated in real-time by other components, the fusion has to periodically make new queries so the decision-making process take into account correct information.

5.2.2 Events

The GUIDE framework has a considerate number of components that have to constantly communicate with each other to make the application receive the correct inputs, change state and deliver the appropriate output. This message exchange is assured by the GBUIF (GUIDE Baseline UI Framework) a bus-based communication system that acts as a publishers/subscriber service. The integration of the GBUIF in the GUIDE framework components is currently in its early stages, however, work has started on elaborating the spec-

ification of the events that are sent and received by every component. The current specification for the GFC events is defined in table 5.1

Event ID	**Pub/Sub**	**Data representation**	**Description**
RawInput	Sub	EMMA	Event sent by input recognizers whenever they detect a new command from the user
AdaptedInput	Sub	EMMA	Event sent by the "Input Adaptation" module whenever it detects a new input from the user
UIRepresentation	Sub	UIML	Event sent by the Dialogue Manager containing the current representation of the application.
CurrentUserModel	Sub	XML	Event sent by the User Model component whenever information about the current user is required
InterpretedCommand	Pub	EMMA	Event sent by the Fusion Module when it interprets the data received from input devices and reaches a decision
CurrentUserModelQuery	Pub	XML	Event sent by the Fusion module to the User Model when it requires additional information about the current user

Table 5.1: Guide fusion core published and subscribed events

The specification of these events is not final, because as the project needs may change, so may its architecture and how messages circulate inside the framework. The upcoming integration of the GBUIF with the GFC will still be based on the events mentioned above and will allow to take the first steps into joining all of the GUIDE framework components that are currently being developed by project partners.

5.2.3 A concrete representation of an interface

The GUIDE Web Browser Interface component is a piece of software that acts as a bridge between the GUIDE framework and the Web browser that is installed in the set-top-box and runs all the Web-based applications. As stated in the previous section, the com-

munication in the GUIDE framework is assured by the GBUIF, including the message exchanging between Web browser and WBI (See Fig. 5.2).

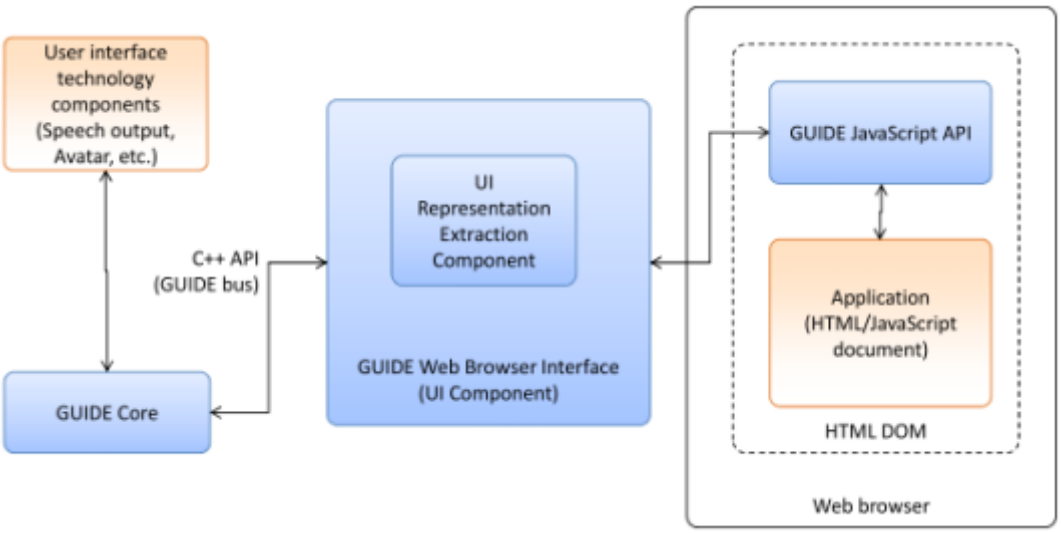


Figure 5.2: WBI Architecture

At any given moment the GUIDE framework is running an application, the fusion module (and other components) must at all times have a sense of the current application state. For the specific case of fusion, the most import data to have is which interactive elements are on screen and their properties. Since the fusion is concerned with input events and the user is capable of expressing an infinite set of actions when interacting, it is important to know how to filter relevant actions for the current context. For this reason, whenever the application executes a change of state, it communicates that change to the WBI through the GUIDE JavaScript API. The WBI possesses a sub-component called the UIREC (User Interface Representation Extraction Component) that is responsible for generating a representation of the user interface. This representation is constructed via UIML (User Interface Markup Language), an XML language oriented for the definition of user interfaces [32], that defines the actual interface elements (e.g. buttons, menus, lists) and their properties. One great feature of this language is its extensibility, due to the fact that the element properties are not defined by the UIML specification, allowing a developer to choose the most adequate properties for their UI controls.

5.2.4 A frame-based approach

The first step of the implementation process was to choose an appropriate fusion scheme or algorithm that would support the needs of the GUIDE fusion core. The most common and well known literature [4] suggests the existence of three main types of architectures for decision-level fusion, which were briefly stated on chapter 2.

The approach chosen to be used in the fusion core was frame-based fusion and it was based in the work of Dumas [4] in the HephaisTK framework fusion engine. In that framework a frame-based algorithm was implemented, however, it does not account for the need to constantly adapt to user and context, which is a primordial task in GUIDE. Despite this fact, the first version of the GFC was designed and implemented with no capacity of adaptation, to test how fusion could work before considering user adaptation, and that is why this frame-based approach as described by Dumas [4] could suffice as an earlier architecture possibility but with a need to evolve over time. This section will focus in the current features of the GFC, that enable the combination of modalities, leaving the details about adaptive strategies to the section dedicated to adaptation (section 5.2.6).

The current implementation uses data structures called frames as the definition above implies. However, the approach used can also be seen as being "hybrid", because it needs to use statistical data to account for user and context data. The following image shows the structure of a typical frame.

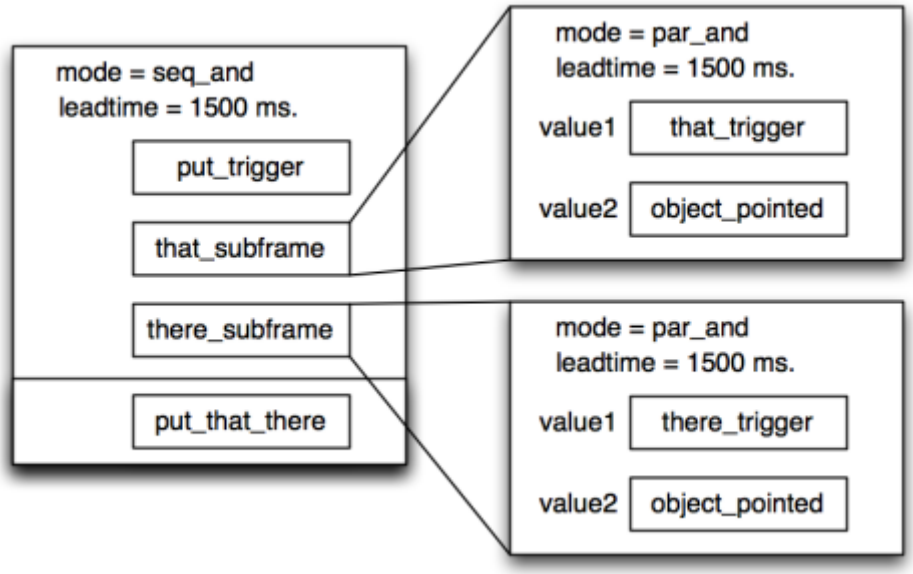


Figure 5.3: GFC Frame Example

The frame structure consists of two major sets. The first one is a set of slots, which can either contain triggers or sub-frames. The triggers are basically conditions that are to be met in order for the slot to be activated, while a sub-frame is a regular frame contained inside another frame, allowing the representation of more complex interaction scenarios. A trigger is associated with one and only one modality (such as speech or pointing) and contains data about the type of modality and the modality relevant token that has to be checked for validation. As an example, it can be seen in the image above, that there is a slot with the name "put trigger" which is a slot of the type "speech" and with token value "put", which means that for it to be activated the user must say the word "put".

The second set of a frame consists of results, which are actions or commands that have to be sent to the dialogue manager once the slots activation occurs. Besides these two data structures within the frame, there are also, at the moment, two attributes that play an important role in the frame activation process. The "mode" attribute defines how the slots are related in time to be activated (i.e the necessary synchronicity of input events). Parallel and sequential triggers are distinguished, as well as coupled (and) and exclusive (or) triggers. Based on these properties, there are four possible values for the "mode" attribute:

- par-and: used when multiple triggers are to be fused together. It is necessary the activation of all triggers (i.e. receive input events that match the trigger condition) for the overall activation of the frame. The order of the received events does not matter, as long as they all arrive in a defined time window.

- seq-and: works as the same way as "par-and" does. The only major difference is that the events that are supposed to validate the slots must be inside the designated time window and must arrive in a pre-defined order so the frame is validated and the results can be sent.

- par-or: describes redundant multimodal triggers having similar meanings. Each one is sufficient for the correct meaning to be extracted, but they all can be expressed at the same time by the user, increasing as such the robustness and recognition rate (e.g. a user issuing a "play" vocal command and simultaneously pushing a play button with the remote control).

- seq-or: to be used when multiple triggers can lead to the same result, but only one of them is to be provided.

For most of the frame synchronization possibilities mentioned above, the attribute "leadtime" is a necessary attribute, that defines the duration of the temporal window in which the slots activation must happen. This is one example of parameter that can be adjusted from user to user according to his profile and that can have a great impact in the interaction. Deciding if a set of events should be fused together or not, is heavily dependent on the time of arrival of events and the designated "leadtime" for the user. Figure 5.4 shows three different cases where the "leadtime" attribute rules if the input events are candidate for fusion. In case A, the events arrive too far apart from each other and so they must be treated as unimodal events (that are not part of a multimodal event). In case B, the events do not overlap in time, but they arrive inside the temporal window defined by the "leadtime" and therefore they have the possibility of being fused together. In case C events arrive inside the same time window and overlap, therefore need for fusion becomes almost evident.

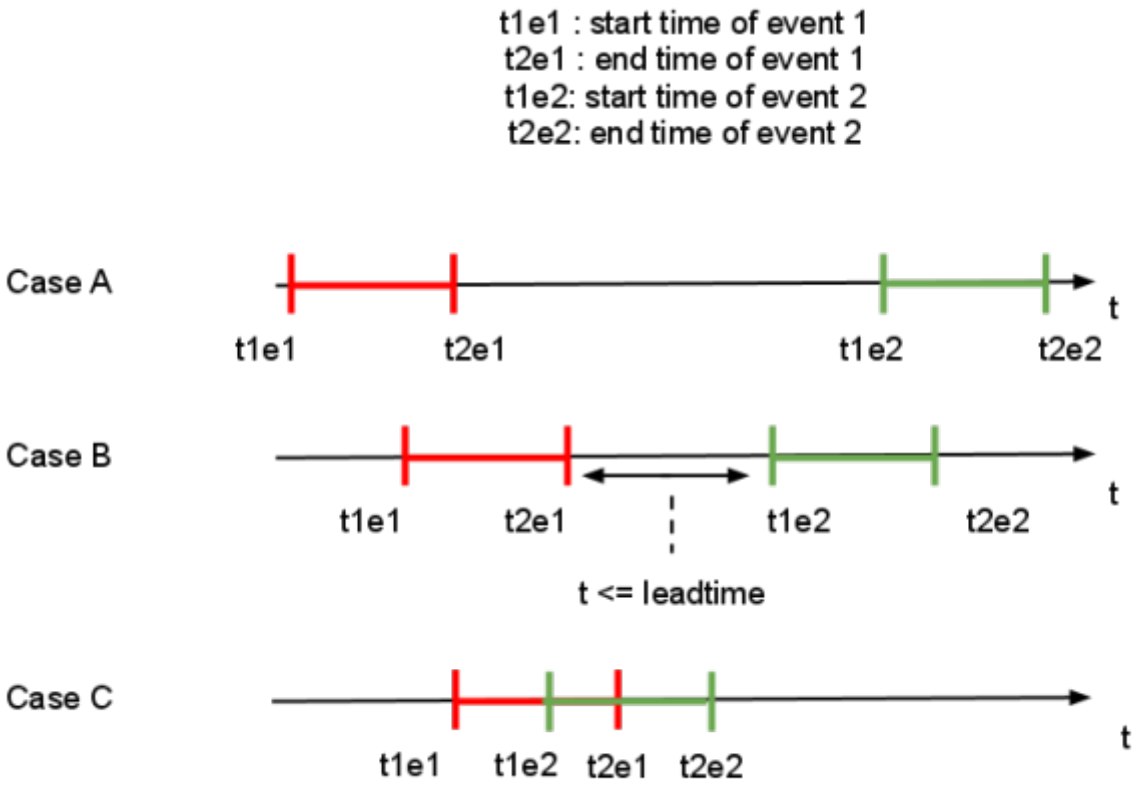


Figure 5.4: Leadtime attribute involved in the fusion of input events

The possibility of using sub-frames inside slots is very important to represent scenarios of interaction that are complex and need to involve more than one mode and different leadtime values. Figure 5.3 shows such an example, specifically of the well known "put that there" paradigm, in which some actions have to occur in parallel and others in sequence.
The next section will discuss how this frame-based implementation copes with the rest of the components that are related to the GFC and the process of frame creation.

5.2.5 Frame creation life cycle

The frame-creation process, exemplified in Figure 5.5 is something that is expected to occur many times during an application life-cycle. As the context of the applications changes (i.e. a state change), the GFC must prepare to potentially receive different type of input events and send the correspondent responses.

As mentioned earlier, the dialogue manager will periodically receive messages from the Web Browser Interface component that contain the representation of the current UI displayed on the screen. This representation is written in an XML based language called UIML (User Interface Markup Language) that allows the definition of the actual interface

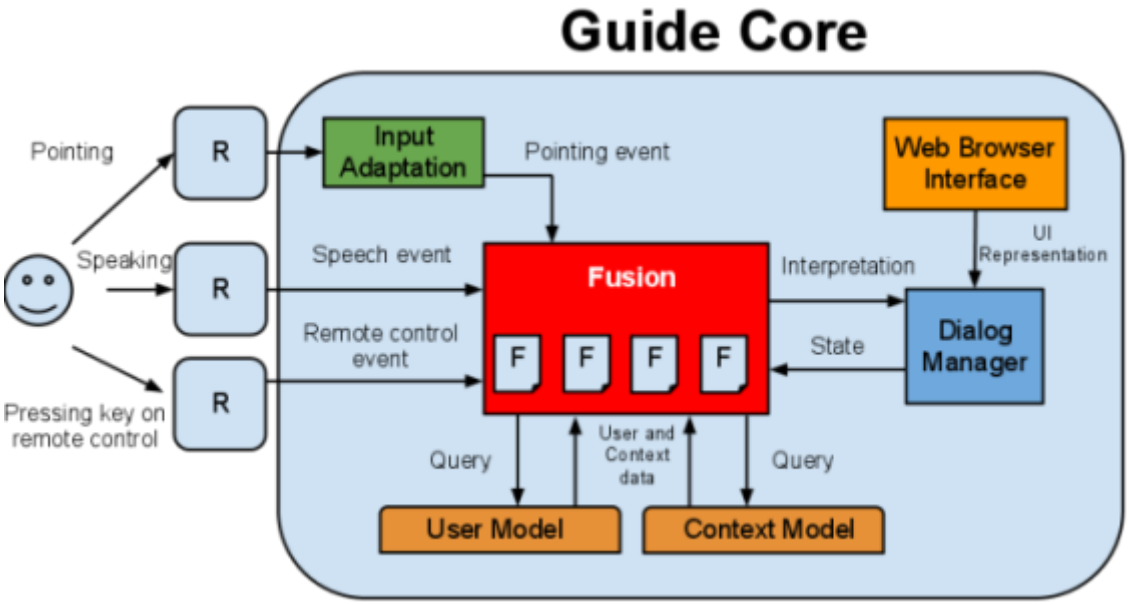


Figure 5.5: GFC Frame Creation Process

elements like buttons, menus, lists and their properties.

When receiving this data, the dialogue manager identifies if and what changes should occur in it, and then sends the current state representation to the fusion core. Once the fusion module has this important piece of information, it fetches additional data by making queries to the user and context model. This information will be used to enhance the frame creation process and imprint adaptive behaviour into it.

When the fusion have both the set of interactive elements on screen and characteristics about user and context, the frame creation process can then begin. For each type of interactive element, specific frames will have to be considered and created. For instance, with buttons, frames will have to be created so that these elements can be clicked using, for instance voice or gestures. Figure 5.6 shows an example of a set of frames that would be created if a button “btn” was present in the current UI representation of an application.

Note that some of the frames exemplified above try to express an abstract view of the input events necessary to trigger the slots. In reality, the slots associated with pointing events for instance, do not contain only one slot but several of them, to check if each coordinate value is above and behind the values it should be (the borders of the button). Most of the choices related to which type of frames to create came from the selection methods thought out for the user trials and the user initialization application. Since the current implementation of the GFC is still in the beginning of its development and for simplicity reasons, the only interactive elements currently being tested are buttons, and because in most Web applications these are the main controls available it was decided that would serve as a good starting point.

Aside from the user model and the UI representation, the other main source of knowl-

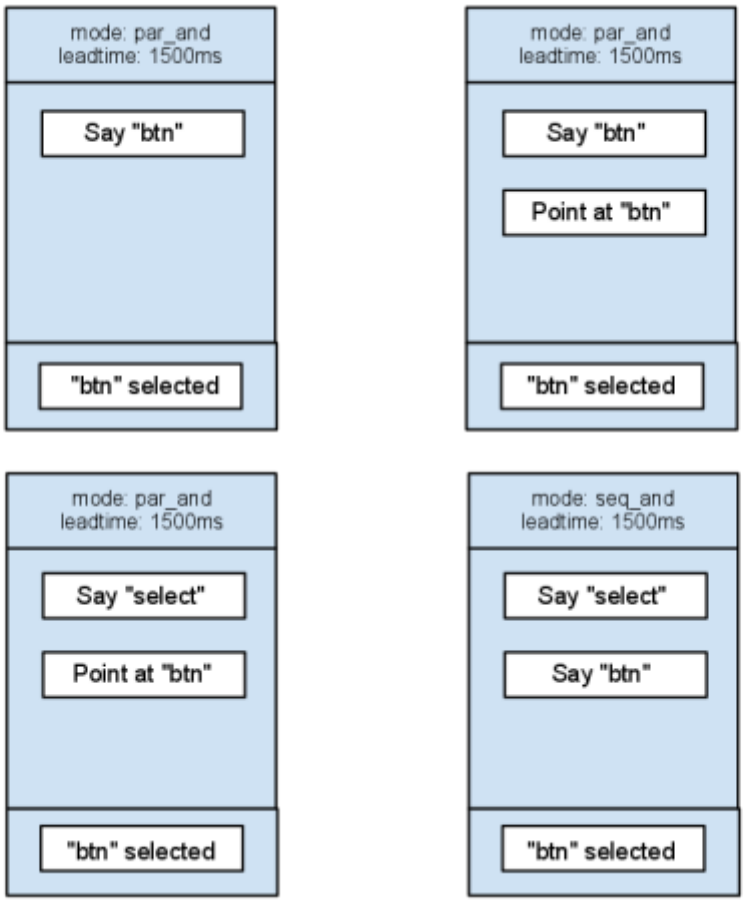


Figure 5.6: Example of frames related to buttons

edge to the fusion core are the recognizers and the input adaptation module. When these components detect a new input from the user, they should process it and send it to the fusion. This processing besides hard-decisions (e.g. the user said hello; she pointed at coordinates x=54, y=56) also support uncertainty (e.g. using confidence score-levels) as well. The input events sent by the recognizers should contain scripts expressed in EMMA [48], a device-independent and multimodal oriented XML language for input that is able to represent this kind of uncertainty. This language was briefly cited in the second chapter of this thesis in the scope of the evaluation of multimodal systems and some of its concepts will be further analysed in the section dedicated to the GFC evaluation.

As input is constantly streamed to the fusion engine, each frame will have to check if its slots are valid or not. Once a frame activates, the fusion sends the corresponding set of results/commands to the dialog manager. This, may or may not trigger a change in the application context which may imply a new state of the application. If this is the case then the whole process will repeat itself and new frames will eventually be created.

Frames are data structures that need to be constantly destroyed and created to reflect the current state of an application. However, since the GUIDE framework runs Web-based

applications for a TV environment, it must also support input related for such an environment. That is the reason why the GFC must create and always have in memory certain frames (denominated "GUIDE frames") that are related to application-independent commands, like for example raising or lowering the volume of TV, turning off the system or an application. This set of commands is not yet properly defined in the project, but some of these ideas were put to practice during the initial evaluations of the GFC.

5.2.6 Adaptation

As suggested throughout this document, there is a great need for adaptation in all of the GUIDE framework, and the GFC is no exception. Since this framework is oriented for making applications accessible to a specific type of users, namely elderly people whom may possess physical or cognitive limitations, and since applications are not usually accessible by their own means, the GUIDE core will have to make sure that they are adequately adapted so users can have an optimal interaction experience.

The fusion that occurs within the GUIDE framework is involved with two different kinds of adaptation, one that happens outside the fusion module and other that happens inside.

- **Adaptation outside the fusion module**

  Besides the self-adaptation of the fusion core, this component also triggers at some point in time, the adaptation of other components in the framework. The following items will state the components which adaptive behaviour can be triggered by actions taken by the GFC.

  - User Model

    Triggering a constant adaptation of the user model is something that is expected from many of the GUIDE framework components. As the interaction between user and system becomes more frequent the former becomes more used to the latter and so their expertise evolves over time, altering their needs towards the system. The disabilities or limitations of the user may also change, meaning that the information captured through the UIA is not longer plausible and therefore changes have to be made in the user model. The GFC in particular, by logging certain interaction patterns can adapt the user model by changing variables such as the "leadtime" of the frames so the system can have a faster response, or so the users can have more time to perform their intended actions.

  - Dialogue Manager

Being the central component of a typical multimodal system, the GUIDE dialogue manager is possibly the component that most sends and receives events involved with adaptation. When the GFC activates one or more frames, the results are sent to the dialogue manager, so it can decide which actions to take next. In most cases sending a result, makes the DM adapt itself to new circumstances and update its current state to the next state in the application logic.

- Recognizers

Recognizers such as regarding speech or gestures must have dictionaries containing the available commands that are to be recognized. Since an application can change its state very often, then the set of commands supported by one of these recognizers at a given time can also differ greatly. For this reason this type of recognizer also have to perform adaptation over-time. The fusion core, when receiving the UI representation from the dialogue manager, realizes which interactive elements are displayed on screen. By sending data, like buttons names to a speech recognizers or a list of supported gestures to another recognizer, they can construct its dictionary adapting to the current context and prepare to receive the upcoming input from the user.

- **Adaptation inside the fusion module**

The main task of the fusion engine of GUIDE is to potentially combine any incoming input from the recognizers, make an interpretation of that data and forward it to the dialogue manager. The key to provide the most suitable interpretation is to take into account critical information that is provided by three main sources: the user model, context model and input events.

- User Model

The User Model component is the one which holds information about the users that interact with the system. These profiles contain data about user limitations or disabilities and are created beforehand, when the user has a first contact with the system and goes through the initialization application. This is perhaps the most important source of information for the fusion module adaptation since the framework main goal is to tackle the accessibility issues of elderly users, that most of time can be very specific to each one of them, and therefore the adaptation for each user will also be very particular.

- Context Model

  The Context Model will provide the fusion module with information about the current context in which the user is inserted and that may influence the decisions made by the system. A noisy or a crowded room are examples of scenarios that affect the performance of the recognizers and consequently of the decisions produced by the fusion.

- Input Events

  Physical events that are produced by users and that are captured by recognizers are the main source of information for a fusion engine, because without input there would not be a need for fusion. It is expected from the recognizers residing outside the GUIDE framework to provide semantic interpretations from a variety of inputs, including but not necessarily limited to, speech, gestures, pointing, remote control and tablet input. This information complemented with the others mentioned above, will enable the fusion core to process and deliver the most trustworthy interpretation.

5.2.6.1 Weight-based adaptation

As previously stated, the current architecture to implement fusion in the GUIDE framework is based on decision-level fusion (because this is type of fusion that supports loosely-coupled modalities) and the specific approach chosen was a frame-based one. However, from what was already described about the frame approach functioning and that is heavily based on Dumas work in the HephaisTK framework [4], it is clear that adaptation to the user and context is not possible with this approach.

The solution considered consists on adopting an hybrid approach where decision-level fusion features are combined with features from opinion-level fusion. Opinion-level fusion by definition [8], occurs when the recognizers no longer provide hard-decisions, but opinions instead. In this context, opinions are characterized by having score-levels, or weights assigned to them, which allows to represent uncertainty in a system (see Figure 5.7).

In the GUIDE framework there are variables and aspects related to the fusion process that can be subject to adaptation, so it is worthy to discuss them. The input events received from the user, related with some modalities, will have weights assigned to them, namely those coming from recognizers susceptible to errors such as the ones that deal with speech and gestures. Other types of interaction such as remote control do not require representation of uncertainty since once an input is detected, although it may have been unintentional, it is certain that has happened. By interpreting score-values the fusion core

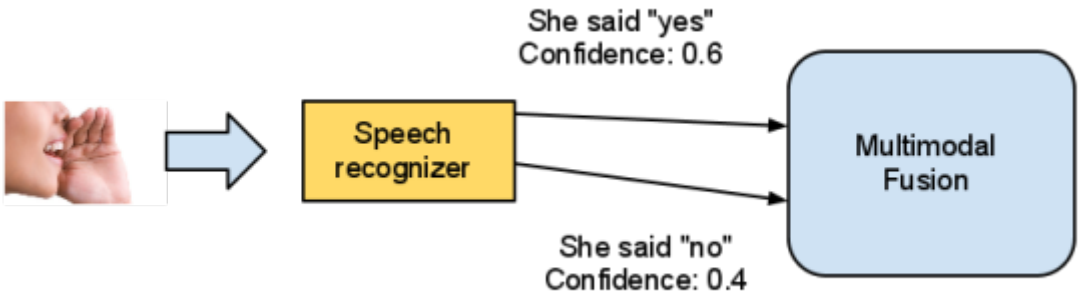


Figure 5.7: Input events based on confidence levels

can decide which frames should preferably be activated over others. Until the semantic data from the recognizers along with these confidence values reach the multimodal fusion module, no adaptation to user occurs. The scoring reflects the reliability of each decision that was produced from the recognizers. However upon arrival, the fusion makes use of the other two sources of information (user model and context model) to change the impact of the confidence values of the inputs, to reach its final decision. The GFC also triggers adaptation on the recognizers, because in certain cases (e.g. speech and gesture recognizers), these external components must form language dictionaries to know what inputs to expect, and so they also have to adapt its behaviour. As the GFC receives new representations of the UI from the dialogue manager, a list of possible speech and gestures commands are forwarded to the correspondent recognizers. The recognizers that will interact with the GUIDE framework, in to order to support uncertainty, or opinions, will produce their output in EMMA, an XML markup language that allows the capture and annotation of data at various stages of the user inputs processing. There are two key aspects associated with the EMMA language: a series of elements (e.g. emma:interpretation, emma:one-of, emma:group) that work as containers for possible interpretations of the user actions, and a series of annotation attributes and elements which are used to produce additional metadata associated with the inputs, such as timestamps (emma:start, emma:end) or confidence values on generated interpretations (emma:confidence). Figure 5.8 shows an example of how an EMMA document can be structured by an air travel reservation system and how application-specific semantics can be inserted (in this case information about flights origin and destination).

Every EMMA document has to possess an emma:emma root element which indicates information about EMMA namespaces, version, and other annotative data. The main container for semantic data is the emma:interpretation element which can work as a wrapper for applications instance data, which will follow or be included in the core part of an EMMA document, a tree of container elements (emma:one-of, emma:group and emma:sequence). In Figure 5.8 an example of the element emma:one-of can be seen. This

```
<emma:emma version="1.0"
    xmlns:emma="http://www.w3.org/2003/04/emma"
    xmlns:xsi="http://www.w3.org/
    2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.w3.org/2003/04/
    emma http://www.w3.org/TR/2009/
    REC-emma-20090210/emma.xsd"
    xmlns="http://www.example.com/example">
  <emma:one-of id="r1"
     emma:medium="acoustic" emma:mode="voice"
     emma:function="dialog" emma:verbal="true"
     emma:start="1241035886246"
     emma:end="1241035889306"
     emma:source="smm:platform=iPhone-2.2.1-5H11"
     emma:signal="smm:file=audio-416120.amr"
     emma:signal-size="4902"
     emma:process="smm:type=asr&version=asr_eng2.4"
     emma:media-type="audio/amr; rate=8000"
     emma:lang="en-US" emma:grammar-ref="gram1"
     emma:model-ref="model1">
   <emma:interpretation id="int1"
    emma:confidence="0.75"
    emma:tokens="flights from boston to denver">
      <flt><orig>Boston</orig>
           <dest>Denver</dest></flt>
    </emma:interpretation>
    <emma:interpretation id="int2"
     emma:confidence="0.68"
     emma:tokens="flights from austin to denver">
      <flt><orig>Austin</orig>
           <dest>Denver</dest></flt>
    </emma:interpretation>
  </emma:one-of>
  <emma:info>
   <session>E50DAE19-79B5-44BA-892D</session>
  </emma:info>
  <emma:grammar id="gram1"
   ref="smm:grammar=flights"/>
  <emma:model id="model1"
   ref="smm:file=flights.xsd"/>
</emma:emma>
```

Figure 5.8: Sample EMMA document (Adapted from [20]

element represent the N-best interpretations that the system came up with at a certain moment and context. Depending on applications logic and requirements, the best and most plausible interpretation will then be chosen among those inside the emma:one-of element. The other two main containers are emma:group for grouping inputs and emma:sequence to represent sequence of inputs in time.

The current implementation of the GFC is capable of parsing EMMA in some degree, and the evaluation script that is currently being used to evaluate the GFC is heavily based on EMMA structure. The scripts used to simulate inputs in the GFC are human-generated and so do not yet use pure EMMA scripts, to keep the process the simplest possible, but they do however keep the main attributes necessary to perform early evaluations such as timestamps (for start and end of an event), type of modality, semantic data and confidence

scores.

Augmenting the GFC with weight-based adaptation proves useful, not only to allow the recognizers to produce outputs that are not absolute truths, thus making them commit less to the decisions made, but also for the GFC, which based on the scores of these input events can make better decisions about frames activation.

5.2.6.2 Implementation

Using a weight-based approach in the GFC has the purpose of improving the overall interaction between user and system, by making use of characteristics contained in user models that reflect the abilities and limitations of users. Most of the variables extracted from the interviews conducted during the user trials gathered information that was mainly relevant for multimodal fission (e.g. button size, inter-spacing, ability to see at distance, see at night, hearing a sound of a certain frequency). The concepts extracted concerning fusion was that users preferred some types of interaction over others. After a final structure of the user model is decided (which is a responsibility of other project partner), the fusion will have to get a mapping of characteristics related to user input capabilities and preferences into scores that represent them. Until this structure is developed, the current GFC implementation assumes that each modality has a score associated with it (ranging from the value of 0 to 1) that reflects the weight that the modality should have, considering the user aptitude for that kind of interaction. For instance, if the “speech” modality posseses a higher weight than the ”gesture“ modality than the input events related to speech are considered more trustworthy. Figure 5.9 shows how the structure of a frame is affected by this new attribute.

With this transformation, frame slots stop being data structures that must receive an exact input to be activated and trigger changes in the application. As mentioned before, by using EMMA the recognizers have a greater flexibility because they output events that are not absolute truths (using the confidence scores) or form different alternatives to what was recognized (e.g. using the “one-of” container). This information combined with the scores now imprinted into slots, reflecting user proficiency with the modalities, allow to leverage frame activation decisions. For the moment, the implementation uses simple rules to make use of both confidence from the inputs and the user model. For instance, if an input event presents a low confidence score in a scenario where information from the user would not be considered, then a slot where that input event fit would not be activated. However, if the information from the user model is used into the decision process and the user presents in the modality related to the slot, a confidence higher than a certain value (e.g. 0.6) then the input is considered more trustworthy, hence activating the slot.

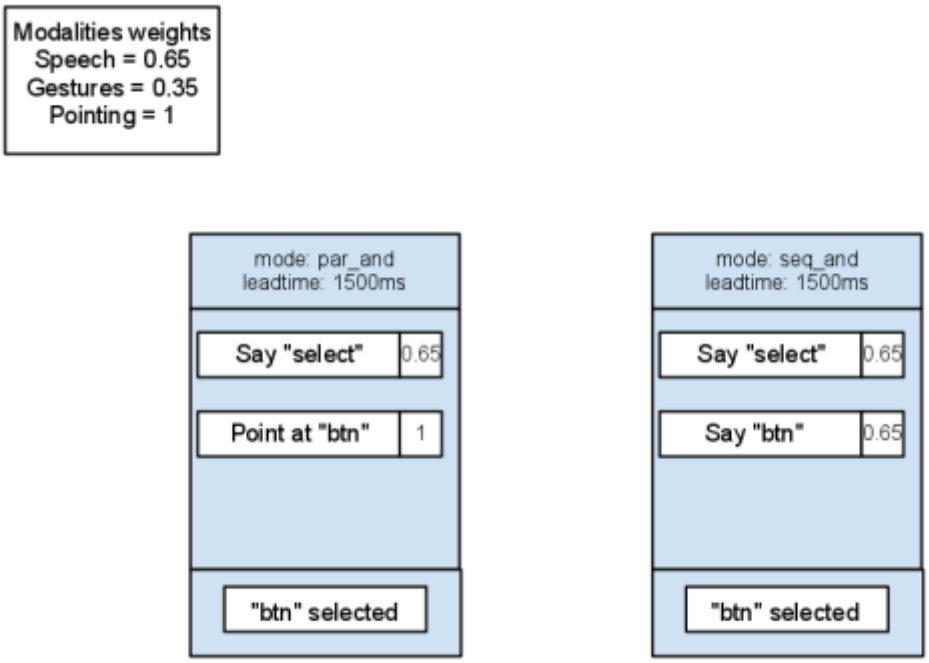


Figure 5.9: Example of frames with slot weights

5.3 Evaluation

The GFC is still in its early development, specially when it comes to adaptation to user. However, a desired approach to be taken throughout the project duration, is to include evaluation practices right from the beginning to assess the state of the module. As the components of the framework, developed by multiple project partners, continue to be integrated with each other, the complexity of the system grows and evaluation, although more accurate for the final product, becomes more complex and troublesome. For all the reasons above, an evaluation framework started to be implemented into the GFC, allowing an early assessment of performance using certain metrics.

A section of the second chapter of this thesis focused on the evaluation of multimodal systems and suggested that errors are not only responsibility of the components themselves but also of the user. When the fusion core does not produce the expected results it may be due to flaws in the algorithms used, low performance of the recognizers or a poorly formed query by the user. In order to cope with the existence of these many variables, the evaluation method implemented, based on the ideas of Dumas [5], allows the simulation of recognizers output, therefore eliminating their error proneness and fully controlling what goes to the fusion engine and when.

At the current state of implementation, in order to run a simulation session, an XML file must be manually created, defining a series of interaction scenarios. Figure 5.10 shows an example of such a file.

```
<?xml version="1.0"?>
<tests>
  <scenarios>
    <scenario>
      <events>
        <event type="speech" token="volume up" start="1000" end="2000" arrival="3000" confidence="0.6" />
        <event type="speech" token="hello" start="5000'" end="7000" arrival="8000" confidence="1" />
      </events>
      <ground-truth>
        <result token="raiseVolume_speech1" />
      </ground-truth>
    </scenario>
    <scenario>
      <events>
        <event type="speech" token="hello" start="100'" end="999" arrival="1000" confidence="1" />
        <event type="gesture" token="wave" start="1000" end="2000" arrival="2000" confidence="1" />
      </events>
      <ground-truth>
        <result token="raiseVolume_speech1" />
      </ground-truth>
    </scenario>
  </scenarios>
</tests>
```

Figure 5.10: Evaluation script example

As it can be seen in the image above, a simulation consists in a series of scenarios. A Scenario is formed by a list of events that are sent one after another to the GFC. These events have attributes reflecting their start and end timestamps, semantic data, type of modality and confidence scores. An interesting feature that can be simulated is the arrival time of the event in the fusion core. Using this data, the evaluation framework is capable of sending the events of a particular scenario over time with defined intervals, thus simulating delays of the recognizers or network. Inserting this kind of feature in the evaluation is extremely important to test situations where the "leadtime" attribute of the frames is to be tested. All time-related attributes are expressed in milliseconds in order to provide broader possibilities in the simulation process. Alongside with the events, the other set that forms a Scenario is the set of ground-truths. The concept of ground-truths in the context of this evaluation framework is the same as the one mentioned in the testbed proposed by Dumas [5], which is defining results that are expected (the ground-truths) to be produced by the fusion engines. At this time, the actual data that is inserted in the XML into the ground-truths, is simply the results that are contained within the frames. As the events are sent to the GFC and frames activated, this produces a set of results that in the end are compared with the pre-established ground-truths.

Figure 5.11 shows a set of use-cases that reflect how a test script could be set up in order to evaluate the impact of the "leadtime" attribute in the user interaction (for the remainder of this example assume that the user presents a "leadtime" of 1500 milliseconds). The three scenarios consist on testing the activation of a frame containing three slots, two of them corresponding to speech commands and one to a pointing event. The result sent upon frame activation is a command meaning that the "news" button has been selected.

```
<?xml version="1.0"?>
<tests>
  <scenarios>
    <scenario>
      <events>
        <event type="speech" token="select" start="1000" end="3000" arrival="3000" confidence="0.6" />
        <event type="speech" token="news" start="4500'" end="5500" arrival="8000" confidence="1" />
      </events>
      <ground-truth>
        <result token="news_button_selected" />
      </ground-truth>
    </scenario>
    <scenario>
      <events>
        <event type="speech" token="select" start="1000" end="3000" arrival="3000" confidence="0.6" />
        <event type="pointing" token="450;450" start="1500'" end="2500" arrival="8000" confidence="1" />
      </events>
      <ground-truth>
        <result token="news_button_selected" />
      </ground-truth>
    </scenario>
    <scenario>
      <events>
        <event type="speech" token="select" start="1000" end="3000" arrival="3000" confidence="0.6" />
        <event type="speech" token="news" start="5000'" end="6000" arrival="8000" confidence="1" />
      </events>
      <ground-truth>

      </ground-truth>
    </scenario>
  </scenarios>
</tests>
```

Figure 5.11: Evaluation use-cases

In the first scenario the events do not intersect in time, but they are, however, inside the temporal window defined by the user "leadtime" (1500 milliseconds) which leads to the activation of the frame. The second scenario exemplifies a situation where the two events intersect each other and when fusion is obviously required. If the fusion engine functions correctly, both of the slots should be activated and correspond to the established ground-truth. On the other hand, the third scenario defines no ground-truth results, which means that in this case no frame should be, in normal circumstances, be activated. This is justified by the fact that the events are too far apart in time, being this distance greater than the user "leadtime". This last case presented is very interesting besides showing how the fusion module should behave itself in such situations, it also shows that it may be beneficial to adopt certain adaptive behaviour over time and not only concerning frame activation. One example of this type of behaviour would be changing the designated user "leadtime" if cases like this are prone to occur often.

The metrics currently being utilized in simulations to measure the fusion engine capabilities are response time (i.e. the time it takes from the moment the GFC receives the last meaningful input event necessary to validate a certain frame and the moment that the results are delivered) and efficiency (i.e. if the set of produced results match the ground-truths that are defined before-hand).

In the future, as more refined adaptation rules concerning frame activation are defined, a new metric should be inserted in the evaluations which is a confidence score. When sending a result or a set of results to the dialogue manager, the fusion core must also provide alongside them their respective confidence-levels. This can open adaptation possibilities for other components such as the dialogue manager, by making it aware of the certainty of the GFC on a given interpretation.

5.4 Conclusions

This chapter introduced the GUIDE fusion module, a component of the GUIDE framework, responsible for the combination of multimodal input. The architecture of the GFC was discussed in order to understand which components of the framework interact with it and their purpose, followed by a description of the events exchanged between them. A small focus was given to describing the WBI component, due to its responsibility of providing the fusion module with representations of user interfaces. These representations are extremely important because they are constructed by the visual representation of the current application state. The GFC has to be aware of the current interactive elements on screen in order to set up interaction scenarios that can be triggered by the user. These scenarios are expressed in data structures called frames. The frame-based approach was used to support the combination of input, by creating frames that once activated, send a list of results or commands to dialogue manager, which then makes decisions based on these interpretations.

Making the fusion process adaptive is one of the major goals of the GFC. By using weight-based adaptation in the interaction process, recognizers can start to provide "opinions" instead of hard decisions, which allow the fusion module to make decisions not only based in "absolute truths". Adaptive features are also applied into the frame data structure to account for the users capabilities concerning each modality available. By making use of these confidence values the GFC can for instance, allow a low-score input event to activate the frame slot and forward an interpretation.

The last section of the chapter discussed the evaluation aspect of the GFC. An evaluation framework is being developed alongside the fusion module, that allows creating simulation scenarios where many variables such as events start/end instant, confidence scores or arrival time can controlled. Another useful aspect of these scenarios is the use of ground-truths, pre-established assumptions of responses that are compared with the actual results produced by the fusion core. The chapter concluded by showing a small number of use-cases involving simulation scenarios and that can affect user adaptation.

Chapter 6

Conclusion

The work developed during this past year span throughout two major periods of the GUIDE project, namely the user requirements phase and the development of framework components. This thesis focused on the development of applications that aided the initial gathering of user requirements and preferences, exemplified how a user model that reflects user impairments can be constructed, and implemented an earlier version of the GUIDE Fusion Core.

6.1 User requirements

One of the major goals of the GUIDE project during this past year was to gather an exhaustive collection of information about users, which included their associated disabilities, limitations and preferences on interacting with multimodal systems. A great part of this information was collected thanks to the existence of the User Trials Application. The development of this application lasted for some months and suffered many iterations during that time. The degree of test customization allowed by the application continuously grew, by enabling the insertion of new interactive elements and types of media. As the tests diversity raise, so did the support for new input devices. Modalities such as pointing recognition, which started by using the Wiimote for interaction (which is not really considered pointing with your fingers) switched to the Microsoft Kinect to allow a more "natural" pointing interaction. Other project partners also provided components for the UTA, such as the animated avatar used in some tests or point recognition software for the Kinect. Integrating all of these external components into the application was a task assigned to FCUL (since we were constructing the UTA) and it proved to be a difficult one at some points due to the diversity of programming languages and communication protocols involved in those components.

Even though the User Trials Application purpose was to be the supporting piece of software for the user trials, it also helped to grasp some of the technicalities of a multimodal system. These included dealing with an application that must possess a dialogue

manager, a central coordinator of the system that besides having to handle the application states, also have to cope with the various input and output channels available. The concept and importance of fusion was already starting to be evident, as most users in the first trials showed a preference for interacting in a multimodal way. The need to implement a complex and robust fusion engine did not emerge, mainly because at this point the application did not have the need for complex cases of interaction or adaptation. Also, the "Wizard-of-Oz" approach adopted, was useful in the sense that by empowering the person running the tests, system flaws such as a lack of speech recognizer could be overcome.

Overall, the User Trials Application was a major asset to the GUIDE project, that resulted in obtaining several observations about user interaction patterns and preferences. The data collected will serve many partners of the project giving them valuable insights about how users behave themselves when confronted with multimodal systems.

Capturing and analysing user requirements is something that is not specific to the prior implementation of the GUIDE framework. As mentioned several times throughout this document, elderly users can be very unique in terms of the limitation and disabilities they present. An application residing within the GUIDE framework, will gather information about these impairments and deliver it to the other components. The User Initialization Application is a prototype of the aforementioned application. Developing this software brought us somewhat close to what applications in GUIDE will look like. The first reason for this was the programming environment, which switched to HTML/JavaScript, the expected application environment supported by the GUIDE framework . The second reason was the implementation of adaptive features, which served to prove why dynamic adaptation can be so important in improving the user experience with the system. One of other objectives in developing this prototype, was to present it in the annual project review. This was done to demonstrate the capabilities and advantages of a system like GUIDE which makes use of adaptation to tackle user accessibility requirements.

6.2 Multimodal Fusion

Developing the GUIDE fusion core was the main task that was planned prior to the beginning of this thesis. However, due to the responsibilities assigned to FCUL, this goal became part of a broader picture. When the development of the GFC began, certain decisions such as the programming language to be used in the implementation of the framework components was not yet very clear. This led to some delays in platform migrations and such. Even though the implementation of the GFC started in a later stage of this thesis work, the contribution given by it is very significant, being one of the main components of the framework, responsible for handling the incoming input from the user and "setting up the scenario" so the rest of components can give continuity to the interaction process. The frame-based approach implemented proved to suffice the current needs of

the GUIDE project, namely in representing a series of interaction scenarios formed from the parsing of a concrete representation of an user interface. Designing, coding and testing this approach, were tasks that received focus for a considerate period of time, leaving less time for the design of adaptive rules. Adaptation to user is one feature that strongly characterizes the GUIDE framework. Part of the adaptive behaviour GFC, as explained in the fifth chapter, is dependent on information provided by other components, such as the user model or the dialogue manager. The majority of these components are also still in an early development phase, and so the adaptation-related information used by the GFC was somewhat inaccurate (e.g. user modalities weights). Despite of this fact, imprinting adaptive data into the frame-based algorithm through some simple rules showed that user interaction can gain from using a weight-based approach. As these rules get more and more refined it is expected to increase even further the capabilities of the GFC and its adaptability to the user.

In later stages of the GUIDE project, it is expected that evaluations will be made, including the whole framework and real users. Until then, it is important to not disregard the importance of evaluating components in an independent fashion. The first year of project was dedicated almost exclusively to testing and assessing user requirements. A similar approach was adopted when developing the GFC, implementing alongside it an evaluation framework capable of test its performance based on certain metrics. In the long run this platform will evolve with the fusion module and be a great asset to it. By running complex simulation scenarios where many variables can be controlled, the GFC can simulate many situations that would be hard to control or debug code-wise if all of the framework was involved. Even if only a subset of the components that interact with the GFC are present, such as input recognizers, there are variables that cannot be easily controlled (e.g. delay on processing, network). These are some of the reasons why the evaluation framework presented in this thesis proves to very valuable, even in this early stages of development.

6.3 Future Work

The GUIDE project, although in its second year of existence, is still in its early development, specially concerning the implementation of the framework components. Each project partner is still developing their components in a separate manner, but preparations for components integration in the framework and between each other has already commenced. As mentioned in chapter five, events are already starting to be defined and will represent the “language” used by components to communicate. This communication will be assured by the GBUIF, a publisher/subscriber messaging framework, responsible for the deliver of events inside the framework. As for the fusion module that has been in development, many improvements in future work can already be foreseen:

- Start the full integration of the GBUIF API in order to enable the communication to and from the GFC. Since events that are either published or subscribed by the fusion module have already been lightly defined, this component can now start to implement ways of sending and receiving these events.

- Once the User Model component has been defined more accurately, studies can be made to understand how to efficiently translate user characteristics contained in these models into weights that are associated with input modalities. As explained in the previous chapter, the current weights that are assigned to each modality (e.g. pointing, speech) to simulate user capabilities are just random values that range from 0 to 1. These values, at the moment, do not translate any kind of user impairments based on a user model. In the later stages of development in the GUIDE framework, it is imperative that the fusion core (as well as the other components) have available a precise characterization of the user. The adaptive behaviour of the current frame approach used to combine input events, is heavily dependent of the user model scores, because these can dictate if a frame slot can be activated.

- The adaptive strategies utilized until now are very basic, not only because of the lack of a proper user model mapping, but also because of the rules used to decide about frame activation. As these rules and algorithms grow in complexity more efficient ways of validating user actions may be found.

  Besides frame activation there are other adaptive features that are worth exploring. The fusion core must not only make adaptation "on-the-fly" like what happens with frame slot weights, but also adopt adaptive behaviour over time. This is an important concept because as the system and applications change, so does the user, and changes to its user model are bound to happen. From the GUIDE framework point of view, the GFC is the first contact of the user with the system and being the first on the line, includes being responsible for perceiving certain aspects of the interaction such as seeing how the velocity or responsiveness of user evolves over time. These are examples of user characteristics that change in time, and that requires an adaptation in the long-term.

- Consideration of other algorithmic approaches to the GFC. The second chapter of this thesis discussed some options available when implementing architectures or algorithms for fusion engines. The frame-based approach was the first choice taken and since it has been proving to present good results other choices were not yet implemented. However, as the project or the multimodal fusion needs may change, other algorithms may be implemented and compared between each other. Choosing which algorithm to perform fusion would also be an interesting feature to be added to the evaluation framework. This does not mean that there is not room for improvement in the frame-based approach. Currently, the frame creation process,

besides GUIDE-related frames, only instantiates frames related to buttons, one of the most common interactive elements in a Web application. Obviously other elements should be addressed and included in the interaction process.

- As the development and integration of each component in the GUIDE framework continues, it is expected that full large scale tests involving all the components shall also occur. However, despite the importance of testing the framework as a whole, the evaluation framework discussed in the fifth chapter of this thesis, will continue to be developed alongside the GFC. Chapter two, discussed an evaluation approach taken in the evaluation of fusion engines, an idea that was adopted in the development of the GFC. By testing the component in an independent fashion, it is much easier to control the variables involved, and therefore obtain more accurate results. For the reasons, a great focus will given to evaluation for the rest of the GFC development period.

Bibliography

[1] Alexandre L. A., Campilho A. C., and Kamel M. On combining classifiers using sum and product rules. *Pattern Recognition Letters*, 22(12):1283–1289, 2001.

[2] Bolt R. A. "put-that-there": Voice and gesture at the graphics interface. *SIGGRAPH 80 Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, 14(3), 1980.

[3] Dumas B., Lalanne D., and Ingold R. Prototyping multimodal interfaces with smuiml modeling language. In *Workshop on User Interface Description Languages for Next Generation User Interfaces CHI 2008*, pages 63–66, 2008.

[4] Dumas B., Lalanne D., and Oviatt S. Multimodal interfaces: A survey of principles, models and frameworks. *Human Machine Interaction*, 5440(2):3–26, 2009.

[5] Dumas B., Ingold R., and Lalanne D. Benchmarking fusion engines of multimodal interactive systems. In *Proceedings of the 2009 international conference on Multimodal interfaces ICMIMLMI 09*, pages 169–176, New York, NY, 2009. ACM Press.

[6] Hartmann B., Abdulla L., Mittal M., and Klemmer S. R. Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition. In *Proceedings of the SIGCHI conference on Human factors in computing systems CHI 07*, page 145, 2007.

[7] Koons D. B., Sparrell C. J., and Thorisson K. R. Integrating simultaneous input from speech, gaze, and hand gestures. In *Intelligent Multimedia interfaces*, pages 257–276. American Association for Artificial Intelligence/The MIT Press, Menlo Park, CA, 1993.

[8] Sanderson C. and Paliwal K. K. Information fusion and person verification using speech & face information. *Research Paper IDIAP-RR 02-33*, 1(33), 2002.

[9] Guide Consortium. Guide: Gentle user interfaces for disabled and elderly citizens. `http://www.guide-project.eu/`.

[10] Fisk A. D., Rogers W. A., Charness N., Czaja S. J., and Sharit J. Designing for older adults. *CRC*, 2004.

[11] Lalanne D., Nigay L., Palanque P., Robinson P., Vanderdonckt J., and Jean-François Ladry. Fusion engines for multimodal input: a survey. In *Proceedings of the 2009 international conference on Multimodal interfaces*, pages 153–160. ACM, 2009.

[12] J. Fozard. *The handbook of the psychology of ageing, chapter Vision and hearing in aging*, pages 150–170. Academic, 1990.

[13] Hauptmann A. G. and McAvinney P. Gesture with speech for graphics manipulation. *International Journal of ManMachine Studies*, 38(2):231–249, 1993.

[14] Neal J. G., Thielman C. Y., Dobes Z., Haller S. M., and Shapiro S. C. Natural language with integrated deictic and graphic gestures. In *Proceedings of the workshop on Speech and Natural Language HLT 89*, pages 410–423, Morristown, NJ, 1989. Association for Computational Linguistics.

[15] Altinçay H. and Demirekler M. An information theoretic framework for weight estimation in the combination of probabilistic classifiers for speaker identification. *Speech Communication*, 30(4):255–272, 2000.

[16] Bouchet J. and Nigay L. Icare: a component-based approach for the design and development of multimodal interfaces. In *CHI04 extended abstracts on Human factors*, pages 1–4. ACM Press, 2004.

[17] Cardoso J. Suporte para interacção multimodal baseada em televisão. Master's thesis, Faculty of Sciences of the University of Lisbon.

[18] Coelho J., Duarte C., Biswas P., and Langdon P. Developing accessible tv applications. ASSETS 2011.

[19] Coutaz J., Nigay L., Salber D., Blandford A., May J., and Young R. M. Four easy pieces for assessing the usability of multimodal interaction: the care properties. In *Proceedings of INTERACT'95*, volume 95, pages 1–7. Chapman & Hall, June 1995.

[20] Johnston J. Building multimodal applications with emma. In *Proceedings of the 2009 international conference on Multimodal interfaces ICMIMLMI 09*, pages 47–54, 2009.

[21] Kittler J., Hatef M., Duin R. P. W., and Matas J. On combining classifiers. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 20, pages 226–239. IEEE Computer Society, 1998.

[22] Ho T. K., Hull J. J., and Srihari S. N. Decision combination in multiple classifier systems. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 16, pages 66–75. IEEE, 1994.

[23] Hall D. L. and Llinas J. Multisensor data fusion. In *Handbook of Multisensor Data Fusion*, pages 1–10. CRC Press, 2001.

[24] Hong L. and Jain A. Integrating faces and fingerprints for personal identification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(12):1295–1307, 1998.

[25] Lawson L., Ahmad-Amr Al-Akkad, Vanderdonckt J., and Macq B. 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*, page 245, New York, NY, 2009. ACM Press.

[26] Nigay L. and Coutaz J. A design space for multimodal systems: concurrent processing and data fusion. In *Proceedings of the INTERCHI 93 Conference on Human Factors in Computing Systems*, pages 172–178, 1993.

[27] Nigay L. and Coutaz J. A generic platform for addressing the multimodal challenge. In *CHI 95 Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 98–105, New York, NY, 1995. ACM Press/Addison-Wesley Publishing Co.

[28] Oviatt S. L. Multimodal interactive maps: Designing for human performance. *Human-Computer Interaction*, 12(1):93–129, 1997.

[29] Oviatt S. L., DeAngeli A., and Kuhn K. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems CHI 97*, volume 97, pages 415–422. ACM Press, 1997.

[30] Oviatt S. L., Cohen P. R., Wu L., Vergo J., Duncan L., Suhm B., Bers J., Holzman T., Winograd T., Landay J., Larson J., and Ferro D. Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions. *Human Computer Interaction*, 15(4):263–322, 2000.

[31] Wu L., Oviatt S., and Cohen P. Multimodal integration-a statistical view. *IEEE Transactions on Multimedia*, 1(4):334–341, 1999.

[32] Abrams M., Phanouriou C., Alan L. Batongbacal, Stephen M. Williams, and Jonathan E. Shuster. Uiml: An appliance-independent xml user interface language. *Computer Networks*.

[33] Johnston M. and Bangalore S. Finite-state multimodal integration and understanding. In *Natural Language Engineering*, volume 11, pages 159–187. Cambridge University Press, June 2005.

[34] Morandell M. M., Hochgatterer A., Fagel S., and Wassertheurer S. Avatars in assistive homes for the elderly. In *Proceedings of the 4th Symposium of the Workgroup Human-Computer interaction and Usability Engineering of the Austrian Computer Society on HCI and Usability For Education and Work*, volume 5298, pages 391–402, Berlin, Heidelberg, November 2008. Springer-Verlag.

[35] Turk M. and Robertson G. Perceptual user interfaces (introduction). *Communications of the ACM*, 43(3):32–34, March 2000.

[36] Krämer N. Freundliche hilfen für abschreckende künstlichkeit? virtuelle agenten für senioren. In *Seniorengerechte Schnittstellen zur Technik Zusammenfassung der Beiträge zum Usability Day VI*, May 2008.

[37] Krämer N., Lurgel I., and Bente G. Emotion and motivation in embodied conversational agents. In *Proceedings of the Symposium "Agents that Want and Like"*, pages 55–61. Hatfield: SSAISB, 2008.

[38] Poh N., Bourlai T., and Kittler J. Multimodal information fusion. In *Multimodal Signal Processing Theory and applications for human computer interaction*, page 153. Academic Press, 2010.

[39] Dragicevic P. and Fekete J-D. Input device selection and interaction configuration with icon. In *People and Computers*, pages 543—-558. Springer Verlag, 2001.

[40] Cohen P. R., Johnston M., McGee D., Oviatt S., Pittman J., Smith I., Chen L., and Clow J. Quickset: multimodal interaction for distributed applications. In *Proceedings of the fifth ACM international conference on Multimedia*, pages 31–40, New York, NY, 1997. ACM.

[41] Gaines B. R. Modeling and forecasting the information sciences. *Inf Sci*, 57-58:3–22, 1991.

[42] Sharma R., Pavlovic V. I., and Huang T. S. Toward multimodal human-computer interface. In *Proceedings of the IEEE 86*, pages 853–869, 1998.

[43] Oviatt S.L. Advances in robust multimodal interface design. *IEEE Computer Graphics and Applications*, 23, September 2003.

[44] Oviatt S.L. Multimodal interfaces. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, chapter 14, pages 286–304. CRC Press, 2nd edition, 2008.

[45] Haenselmann T. Foreword to the special issue on multimedia sensor fusion. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 6(4), November 2010.

[46] Schlomer T., Poppinga B., Henze N., and Boll S. Gesture recognition with a wii controller. In *Proceedings of the 2nd international conference on Tangible and embedded interaction TEI 08*, page 11, New York, NY, 2008. ACM Press.

[47] Wahlster W. User and discourse models for multimodal communication. In *Hewlett Packard Symposium on Artificial Intelligence*, pages 45–67. ACM Press, New York, NY, 1991.

[48] W3C. Emma: Extensible multimodal annotation markup language: W3c recommendation. `http://www.w3.org/TR/emma/`.