



Vrije Universiteit Brussel

Faculty of Science
Department of Computer Science
The Web & Information Systems Engineering
(WISE) Laboratory

Mobile Multimodal Interaction: An Investigation and Implementation of Context-dependent Adaptation

Graduation thesis submitted in partial fulfillment of the
requirements for the degree of Master in Computer Science

Maria Solorzano

Promoter: Prof. Dr. Beat Signer
Advisor: Dr. Bruno Dumas

August 2012



Acknowledgements

After finishing this journey, I would like to sincerely thank all the people that walked next to me and helped me to achieve this goal.

First, I would like to express my gratitude to both my promoter, Prof. Dr. Beat Signer, and my supervisor, Dr. Bruno Dumas, for their unconditional support throughout the development of this thesis. Thank you very much for always being available for any discussion, for your quick answers and good advice. All the ideas, suggestions and remarks you pointed out during the different meetings definitely guided me and helped me out. Vielen herzlichen Dank Prof. Signer!, Merci beaucoup Bruno!

I also would like to thank my friend Gonzalo, for his encouragement and help during difficult times. Finally, I would like to thank my parents, sister and boyfriend for their amazing support and love. You are the motor that always keep me going.

Abstract

Over the last ten years, the use of mobile devices has increased drastically. However, mobile users are still confronted with a number of limitations imposed by mobile devices or the environment. The use of multimodal interaction in mobile interfaces is one way to address these limitations by offering users multiple alternative input modalities while interacting with a mobile application. In this way, users have the freedom to select the input modality they feel most comfortable with. Furthermore, the intelligent and automatic selection of the most suitable modality according to changes in the context of use is a subject of interest and continuous study in the field of mobile multimodal interaction.

There exist different surveys and systematic studies providing an overview of context awareness, multimodal interaction as well as adaptive user interfaces. However, they are all independent surveys and do not provide a unified overview over context-aware adaptation in multimodal mobile settings. A main contribution of this thesis is a detailed investigation and analysis of the state of the art in mobile multimodal interaction with a special focus on context-dependent adaptation. The presented study covers the research in this domain over the last 10 years and we introduce a classification scheme based on relevant concepts from the three related fields. In addition, based on the analysis of existing research, we propose a set of guidelines targeting the design of context-aware adaptive multimodal interfaces. Last but not least, we assess these guidelines and explore our study findings by designing and implementing the Adaptive Multimodal Agenda application.

Contents

1	Introduction	1
1.1	Context	1
1.2	Problem Definition and Justification	3
1.3	Research Objectives and Approach	3
1.4	Thesis Outline	4
2	Background Studies	6
2.1	Post WIMP Interfaces	6
2.2	Multimodal Interaction	11
2.2.1	Characteristics	11
2.2.2	Fusion and Fission	13
2.2.3	CARE Properties	16
2.3	Mobile Interaction	17
2.3.1	Characteristics	17
2.3.2	Mobile Devices	20
2.3.3	Context Awareness	23
2.4	Adaptive Interfaces	25
2.4.1	Characteristics	25
2.4.2	Conceptual Models and Frameworks	28
2.4.3	Adaptivity in Mobile and Multimodal Interfaces	31
3	An Investigation of Mobile Multimodal Adaptation	33
3.1	Objectives and Scope of the Study	33
3.2	Study Parameters	35
3.3	Articles Included in the Study	37
3.3.1	User-Induced Adaptation	37
3.3.2	System-Induced Adaptation	43
3.4	Analysis	47
3.4.1	Combination of Modalities	48
3.4.2	Context Influence	52
3.4.3	System-Induced Adaptation	56

3.5	Guidelines for Effective Automatic Input Adaptation	60
4	Analysis, Design and Implementation of an Adaptive Multimodal Agenda	63
4.1	Motivation	63
4.2	Analysis and Design	64
4.2.1	Context and Modality Suitability Analysis	65
4.2.2	Multimodal Task Definition	66
4.2.3	Adaptation Design	67
4.3	Architecture	69
4.4	Technology	71
4.4.1	Android	71
4.4.2	Near Field Communication	73
4.5	Implementation	74
4.5.1	Views and Activities	76
4.5.2	Recognition of Input Modalities	77
4.5.3	The Multimodal Controller and Fusion Manager	83
4.5.4	The Context Controller and Policy Manager	87
4.5.5	Summary	90
5	Conclusions and Future Work	91
5.1	Summary	91
5.2	Future Work	92

List of Figures

2.1	Comparison of two desktop computers over twenty years	7
2.2	Multimodal architecture	14
2.3	Different levels of fusion	15
2.4	Three layers design guideline for mobile applications	21
2.5	Mobile terminals taxonomy	22
2.6	Built-in mobile sensors	23
2.7	Adaptation spectrum	26
2.8	Adaptation process: agents and stages	29
2.9	Adaptation decomposition model	30
3.1	Scope of the study	35
4.1	Three step process for creating a calendar event	64
4.2	Top level architecture	70
4.3	Android stack	72
4.4	NFC products	74
4.5	Ndef record	74
4.6	Android-based implementation of the top level architecture	75
4.7	User interface	77
4.8	EventOfInterest class and subtypes	78
4.9	NFC calendar events	79
4.10	Acceleration readings while executing <i>left</i> and <i>right</i> flick gestures	81
4.11	Acceleration readings while executing <i>back</i> and <i>forward</i> flick ges- tures	81
4.12	Acceleration readings when executing the <i>shake</i> gesture	82
4.13	Recognised gestures	82
4.14	The <code>MultimodalController</code>	83
4.15	Fusion Manager classes	84
4.16	Context frame	86
4.17	No matching slot	86
4.18	Slot match	87
4.19	<code>ContextController</code>	87

4.20 Suitable modalities for the <i>indoor</i> location and different <i>noise level</i> values	89
4.21 Suitable modalities for <i>outdoors</i> location and different <i>noise level</i> values	90

List of Tables

2.1	Context implications in perceptual, motor and cognitive levels . . .	19
3.1	User-induced adaptation in mobile multimodal systems	38
3.2	System-induced adaptation in mobile multimodal systems	44
3.3	Modalities combination summary	50
3.4	Modality suitability based on environmental conditions	53
3.5	System-induced adaptation core features	56
4.1	Context analysis	65
4.2	Ease of use of different input modalities according to context . . .	66
4.3	Supported input modalities and interaction techniques	67
4.4	Indoor locations: supported input modalities	68
4.5	Outdoors locations: supported input modalities	69

Chapter 1

Introduction

1.1 Context

Over the past decade, the usage of mobile devices has increased exponentially, as can be seen from the statistics showing how mobile sales all over the world have dramatically increased from the year 1998 to our days [4, 5]. Mobile devices were originally conceived just as an extension of the conventional telephone by providing communication on the go. However, due to the fast development of technology and the pervasive presence of Internet connection in our time, these devices have become increasingly multifunctional. Nowadays, they provide a wide set of functionality besides their original purpose and users are able to perform everyday tasks using one single device.

A lot of academic research has been done in the *mobile computing* field, specifically addressing the inherent limitations of mobile devices, such as small screen size, limited memory, battery life, processing power and network connectivity. These hardware limitations affect the usability of the applications as well. Hence, novel and new interaction modes have been explored to cope with mobile usability problems. One particular area of interest in this field is *mobile multimodal interaction*. This topic is closely related to relevant research areas that have been widely studied, namely *multimodal interfaces* and *mobile interaction*.

Human communication is naturally multimodal, involving the simultaneous interaction of modalities such as speech, facial expressions, hand gestures and body postures to perform a task [15]. A multimodal interface combines multiple input or output modalities in the same interface, thereby allowing the user to interact in a more natural way with the device. These modalities refer to the multiple ways in which a user can interact with the system.

Diverse studies in this area have shown different possibilities in which modalities can be combined, for instance the pioneering and well known Bolt's "Put that There" system [13]. In his work, hand gestures and speech are used in a complementary fashion, allowing the users to move objects exhibited on a wall display. For example, the voice command "*Put that there*" is accompanied with 2 synchronised hand gestures that indicate the object that is going to be moved and its final position.

Moreover, one task can be performed in different ways using equivalent modalities. For example, in the application presented by Serrano et al. [100] it was possible to fill a form's text field either by typing the text with the keyboard or by speaking a word. Users can select which mode of interaction better fits the task they are performing depending on their current context. According to Oviatt et al. [80], error handling and reliability are improved in this way.

Nonetheless, multiple topics are the subjects of continuous research effort in the field, for instance the modality conflict resolution or the intelligent adaptation of input/output modalities based on contextual information.

Furthermore, multimodal systems can be hosted on small portable devices and mobile interaction studies are used as guidelines to decide how different modalities can be combined in the mobile setting. The context in which mobile users interact with their devices is totally different to the traditional desktop environment. Users are exposed to perceptual, motor, social and cognitive changes as stated by Chittaro et al. [19].

Studies related to Mobile HCI have proposed new interaction styles to deal with these constraints. Current work in the field explores how to facilitate mobile interaction using novel interaction initiatives such as mobile gestures (shaking or tilting the device), contactless gestures (swiping the hand in front of the screen device) or real world object communication (approaching the device to rfid tagged real world objects). In the same way, the use of context information to automate and reduce a user's cognitive load is an area of continuous research in this field.

1.2 Problem Definition and Justification

The potential of multimodal interaction in the specific setting of mobile interaction has not been thoroughly explored. Several approaches and initiatives have been described in diverse papers but to date, few have summarized in a systematic way these findings.

There are extensive studies and surveys in regards to multimodal interfaces as a general field of study [32, 49, 33]. In these studies a thorough analysis of models, architectures, fusion and fission algorithms and guidelines is presented. However, no single study has surveyed the possible combinations of modalities when considering mobile devices and change of context. Therefore, new practitioners and researchers face a steep learning curve when entering this novel field.

In consequence, the need for a systematic and comprehensive study that surveys the state of the art in mobile multimodal interaction field is evident. Therefore, the present thesis presents a study that reviews and categorizes prominent research work in the field and comes out with guidelines that facilitate the design of mobile multimodal applications. Such a survey study could be used as a starting reference for anyone interested in conducting research in this field. Furthermore, promising and underexplored areas are identified and used as a basis for further research work.

1.3 Research Objectives and Approach

The main goal of this work is to conduct a survey on mobile multimodal interaction. This survey has as main objective to analyse existing work considering mobile devices solutions which use different modalities as input channels. In particular, the goal is to review research work where the input modality selection either induced by the system or the user is influenced by environmental changes.

The expected outcomes of this research work are:

- ▷ A systematic study that fulfils three specific research objectives. Namely, a categorisation of prominent research work, a thorough analysis of reviewed articles in terms of composition and adaptation level as well as in terms of environment influence. And, last but not least, the presentation of a set of design guidelines.
- ▷ A proof of concept application based on the study findings.

Under this scope and to fulfil the goals of the project, the workflow has been divided in three main phases. In the first phase, a review of the state of the art in the related research fields is conducted. The core concepts and characteristics from each field are thoroughly studied with the objective of distinguishing important features that can be further used in the study. Additionally, during this phase the selection of the articles that are going to form part of the study is performed.

The second phase of this thesis focusses on the establishment of the study parameters and the classification of the selected articles in recapitulative tables. Using this information, a three level analysis (modality composition, context influence, system induced adaptation) is performed. At the end, a set of guidelines are define in consideration of findings from the study and also existing guidelines from the related research fields.

Finally, in the third phase, a proof-of-concept multimodal application on a smartphone running the Android operating system is implemented based on the study findings.

1.4 Thesis Outline

This thesis is structured in 4 chapters. The remaining chapters are distributed as follows:

Chapter 2 describes the state of the art in multimodal interaction, mobile interaction and adaptive interfaces. For each research field the formal definition, a description of the main characteristics, the perceived end-user benefits and existing design guidelines are presented. Additionally, the core concepts related to each field are reviewed as well. For instance, the multimodal interaction section covers the description of topics such as multimodal fusion, fission and the CARE model. On the other hand, the section devoted to mobile interaction, describes a mobile device taxonomy and addresses the mobile paradigm of context-awareness. Finally, in regards to adaptive interfaces, models and frameworks that formalize the adaptation process are presented.

Chapter 3 describes the survey study on mobile multimodal adaptation. The chapter begins by giving the motivation, objectives and scope of the study. Next, the study parameters as well as a description of the related work is presented. Furthermore, a dedicated section addresses the analysis of the previously classified information. The chapter ends with a description of the design guidelines.

The development of the proof of concept application is the central topic of chapter 4. The chapter begins by describing the motivation and proposes an application that supports the use of multiple modalities in different mobile contexts. Based on the proposed application, the analysis and design phases are described. It is worth mentioning that the design phase relies on the usage of the proposed guidelines. Then, a detailed description about the architecture, technology and implementation details are provided as well.

Chapter 5 presents some conclusions and lists a number of possibilities for future work.

Chapter 2

Background Studies

Interfaces are the medium by which humans interact with computer systems. Each type of interface comprises specific characteristics and imposes features and constraints that characterize all the manners in which a user can interact with the computer. These specific forms of man-machine communication are known as interaction styles.

This thesis particularly focuses in research areas related to multimodal and mobile interfaces. Therefore, the current chapter provides the necessary conceptual background related to these fields. First, an overall overview of the history, characteristics and examples of the next generation of interfaces styles is presented. Subsequently, main concepts, features and characteristics as well as the benefits from multimodal interaction, mobile interaction and adaptive interfaces are described in detail.

2.1 Post WIMP Interfaces

Interface styles have evolved from the command line type of interface introduced in the early 50's, only used by expert users, to WIMP interfaces, which refer to the windows, icons, menus and pointer interaction paradigm. The WIMP paradigm was introduced of 1970 at Xerox Parc, widely commercialised by Apple in the 80's and is until nowadays the de facto interaction style among desktop computers.

Surprisingly, it can be seen that the changes in the interaction styles paradigms did not occur very fast. As stated by Van Daam [109], the changes that have been observed in the past 50 years in terms of interaction styles are not as dramatic as the yearly changes observed in hardware technology. Beaudouin-Lafon [11]

showed and demonstrated how in twenty years the same personal desktop computer varied considerably in price and hardware specifications but highlighted that the graphical user interface remained the same over the years. Figure 2.1 illustrates this comparison.

Three factors were highlighted as the main reasons that turned WIMP interface style in the GUI standard [109], namely: the relative easiness of learn and use, the ease of transfer knowledge gained from using one application to another because of the consistency in the look and feel and the capability of satisfying heterogeneous types of users.

			
	original Macintosh	iMac 20	comparison
date	January 1984	November 2003	+ 20 years
price	\$2,500	\$2,200	x 0.9
CPU	68000 Motorola 8 MHz 0.7 MIPS	G5 1.26 GHz 2250 MIPS	x 156 x 3124
memory	128KB	256MB	x 2000
storage	400KB floppy drive	80GB hard drive	x 200000
monitor	9" black & white 512 x 342 68 dpi	20" color 1680 x 1050 100 dpi	x 2.2 x 10 x 1.5
devices	mouse keyboard	mouse keyboard	same same
GUI	desktop WIMP	desktop WIMP	same

Figure 2.1: Comparison of two desktop computers over twenty years. Image taken from [11]

Although the acceptance of WIMP interfaces among users is evident and indisputable, HCI researchers have analysed their weaknesses and limitations in several studies [109, 41]. According to Turk [107], the GUI style of interaction, especially with its reliance on the keyboard and mouse, will not scale to fit future HCI needs. Most computers limit the number of input mechanisms to these peripheral devices, hence restricting the number and type of user actions to typing text or performing a limited set of actions using special keys and the mouse. Furthermore, the ease of use of WIMP interfaces is affected when the complexity of an application increases. Users get frustrated spending too much time manipulat-

ing different layers of GUI components to perform a task. Finally, today's devices offer touch screens, embedded sensors, as well as high resolution cameras and this hardware technology also demands a different mode of interaction. A summary of the advantages and disadvantages of WIMP Interfaces is listed below.

Advantages

- ▷ Easy to use
- ▷ Easy to learn and adopt
- ▷ Targeted to heterogeneous types of users
- ▷ Very efficient for office tasks

Disadvantages

- ▷ Becomes difficult to use when the application becomes bigger and more complex
- ▷ Too much time is spent on manipulating the interface instead of the application
- ▷ Mapping between 3D tasks and 2D control is much less natural
- ▷ Mousing and keyboarding are not suited for all users
- ▷ Do not take advantage of multiple sensory channels communication
- ▷ The interaction is one channel at a time, input processing is sequential

These shortcomings served as driving force to explore and study new alternatives and solutions. Since approximately the year 2000, the *next generation of interfaces* [73] have seen the light. New types of interfaces and interaction styles have been explored, these interfaces do not rely on the direct manipulation paradigm and seek that users achieve an effective and more natural interaction with the computer. Formally, this type of interfaces are known as post-WIMP interfaces. As defined by Van Damme [109], a post-WIMP interface contains at least one interaction technique that does not depend on the classical 2D widgets such as menus and icons.

As mentioned in [48, 95, 47], representative examples of this new type of interfaces and interaction styles are:

- ▷ Virtual, mixed and augmented reality [71, 114]: virtual reality refers to a type of environment in which the user is totally immersed and able to interact with a digital and artificial world. Sometimes this world resembles the reality but it also can recreate a world that does not necessary follow physics laws. Globes and head mounted displays are used as input interaction devices. Augmented reality on the other hand, refers to the environment in which real objects are mixed with virtual objects. For instance, El Choubassi et al. [35] present an augment reality- based tourist guide, that allows users to select a point of interest with the cellular phone camera and then the system augments the image with additional digital content like photos, links or review comments. Finally, mixed reality refers to an environment where reality and digital objects appear at the same time within a single display.
- ▷ Ubiquitous computing [113]: the main goal behind this interaction paradigm is that computing should disappear into the background so that users can use it according to the task that they are performing at the current moment. Weiser [113] envisioned it as: “*machines that fit the human environment instead of forcing humans to enter theirs*”. Technologies like embedded systems, RFID tags, handheld devices are enabling to achieve a pervasive computing environment.
- ▷ Mobile Interfaces: mobile computing is a paradigm where computing devices are expected to be transported by the users during their daily activities. Due to this mobility factor, mobile interfaces have small screens and a restricted number of keys and controls. Mobile interfaces introduced novel input techniques that were not known in desktop computers, for instance trackballs, touchscreens, keyboards or cameras.
- ▷ Multitouch and Surface Computing [99]: current research has presented new kinds of collaborative touch-based interactions that use interactive surfaces as interface. These interfaces allow multi-hand manipulations and touching possibilities as well as improve social usage patterns.
- ▷ Tangible User Interfaces [46]: a TUI allows users to interact with digital information through the physical environment by taking advantage of the natural physical affordance of everyday objects.
- ▷ Multimodal Interfaces [13, 80]: a MUI allows users to combine two or more input modalities in a meaningful and synchronised fashion with multimedia output. These interfaces can be deployed on desktop as well on mobile devices.

- ▷ Attentive Interfaces [111]: a AUI measures the user's visual attention level and adapts the user interface accordingly. According to Vertegaal [111] by statistically modelling attention and other interactive behaviours of users, the system may establish the urgency of information or actions they present in the context of the current activity.
- ▷ Brain Computer Interfaces [75]: in these interfaces, humans intentionally manipulate their brain activity in order to directly control a computer or physical protheses. The ability to communicate and control devices with thought alone has especially high impact for individuals with reduced capabilities for muscular response.

This plethora of interface styles aims to make the interaction with the system more natural. Their common goal is that users develop a more direct communication with the system by allowing them to use actions that correspond to the everyday practice in the real world. As stated by Turk [107], naturalness, intuitiveness, adaptiveness and unobtrusiveness are common properties from this type of interfaces.

According to Jacob et al. [48] these new interface styles that were studied independently from each other do share similar characteristics. Based on this affirmation, the authors described a conceptual framework called Reality-Based Interaction (RBI). The framework allows to unify the emerging interface styles under one common concept. It relies on user's pre-existing knowledge of the daily physical world and is built upon four main principles:

- ▷ Naïve Physics: refers to the human perception of basic physical principles, hence interfaces simulate properties from the physical world like gravity or velocity. For instance, tangible interfaces may use the constraints that everyday objects impose to suggest to users how they should interact with the interface.
- ▷ Body Awareness and Skills: refers to the knowledge that a person has of their own body and movement coordination. For example, mobile interfaces hosted on smartphones take this aspect in consideration when the user puts the phone near to their ear and the device screen gets blocked.
- ▷ Environment Awareness and Skills: refers to the sense that people have of their surroundings as well as the skills that they develop to interact within their environment. For instance, attentive interfaces and mobile context aware interfaces might use environmental properties like the noise level to change the interface or content accordingly.

- ▷ Social Awareness and Skills: refers to the awareness that people have of the persons surrounding them. This capability leads to develop skills to interact with them. For example, using interactive surfaces like the Microsoft Surface, users are aware of the presence of others and collaborate with each other to achieve a task.

2.2 Multimodal Interaction

Previously, it was highlighted that one of the weaknesses of the WIMP interface is its unimodal type of communication. In everyday communication, the combination of different input channels is used to increase the expressive power of the language.

The adaptation of this behaviour in the digital world was first observed in 1980, when Bolt [13] introduced the concept of multimodal interfaces and presented the “Put that there” system. From then on, the field has expanded rapidly and researchers have investigated models, architectures and frameworks that allow to design and implement systems that support multiple and concurrent input events.

2.2.1 Characteristics

The definition of what a multimodal interface or system is, does not vary considerably between different authors. All convey to say that a multimodal system allows to process two or more input and output modalities in a meaningful and synchronised manner. Oviatt [83] describes such systems as follows:

Multimodal systems process two or more combined user input modes such as speech, pen, touch, manual gestures, or gaze in a coordinated manner with multi-media system output.

The different input modes are also referred in this context as interaction modalities. Nigay et al. [74] described an interaction modality as the coupling of a *physical device* \mathbf{d} with an *interaction language* \mathbf{L} :

$$\text{im} = \langle \mathbf{d}, \mathbf{L} \rangle.$$

The *physical device* comprises the sensor or part of hardware that captures the input stream emitted from the user, for example a mouse or microphone. The *interaction language* refers to the set of well-formed expressions that convey a

meaning, in other words the interaction technique that is used. For instance, pseudo natural language and voice commands are both interaction languages for the speech modality. Thus, the interaction modality *Speech* can be formally described as the couple <microphone, pseudo natural language> or <microphone, voice commands>.

Dumas et al. [32] highlighted that two main features distinguished this type of interaction and systems from others, namely:

- ▷ **Fusion of different type of data:** This type of systems should be able to interact with heterogeneous and simultaneous input sources, thus be able to perform parallel processing in order to interpret different user actions. From an interaction point of view, these interfaces allow users to perform redundant, complementary and equivalent input events to achieve a task.
- ▷ **Real-time processing and temporal constraints:** The effective interpretation of the multiple input and output events depends on time synchronised parallel processing.

The main benefits of these type of interfaces for users are twofold:

- ▷ **Error Handling:** According to Oviatt [80], these types of interfaces possess a superior error handling capability. Studies found mutual disambiguation and error suppression ranging between 19 and 41 percent [79]. Error handling refers to error avoidance and to a better error recovery capability. The author argued that users have a strong tendency to switch modalities after system recognition errors.
- ▷ **Flexibility:** A well-designed multimodal system gives users the freedom to choose the modality that they feel best matches the requirements of the task at hand. Additionally, according to Oviatt et al. [82], multiple modalities allow to satisfy a wider range of users, tasks and environmental situations.

Handling multiple input and output modalities adds complexity during the design and development phase. Therefore, guidelines to design a usable and efficient multimodal interface have been addressed by different authors. Reeves et al. [87] exposed six core features that should be taken into consideration, namely:

MU-G1 *Requirements Specification:* Besides the traditional requirements gathering process, designers should target their applications for a broader range of users and contexts of use.

- MU-G2 *Multimodal Input and Output*: In order to provide the best modality or combination of modalities, it is important to take into account cognitive science literature. This foundations principles allow to maximise the advantages of each modality, in this way reducing a user's memory load in certain tasks and situations.
- MU-G3 *Adaptivity*: Multimodal interfaces should adapt to the needs and abilities of different users, as well as different contexts of use, for instance by disabling speech input mode in noisy environments.
- MU-G4 *Consistency*: Input and output modalities should maintain consistency across the whole application. Even if a task is performed by different input modalities, the presentation should be the same for the user.
- MU-G5 *Feedback*: The current status must be visible and intuitive for users. In this context, the status refers to the input and output modalities that are available to use at any moment.
- MU-G6 *Error Prevention/Handling*: To achieve better error prevention or correction rates, the interface should provide complementary modalities to perform the same task. In this way, users can select the one that they feel that is less error prone.

2.2.2 Fusion and Fission

According to Dumas et al. [32] a multimodal application consist of four main components which are depicted in Figure 2.2. First, the *Modalities Recognizers* are in charge of processing the sensor's data or capture the different types of user events. Then, this raw information is sent to a component called *Fusion Manager*. This component is the heart of a multimodal system, since it is in charge of capturing the diverse events and providing an interpretation that has a semantic meaning for the domain of the running application. For instance, if e_1 and e_2 are two events fired by an user, the order in which these events are executed may lead to a totally different output from this component. The output obtained by the fusion manager is received and processed by the *Dialog Manager*. This component is in charge of sending a specific GUI action message based on the fusion manager decision, the status of the application and the current context. This GUI action message may first be processed by another important component called the *Fission Manager*. This component is in charge of selecting the best output modality according to the following parameters: context, user model and history.

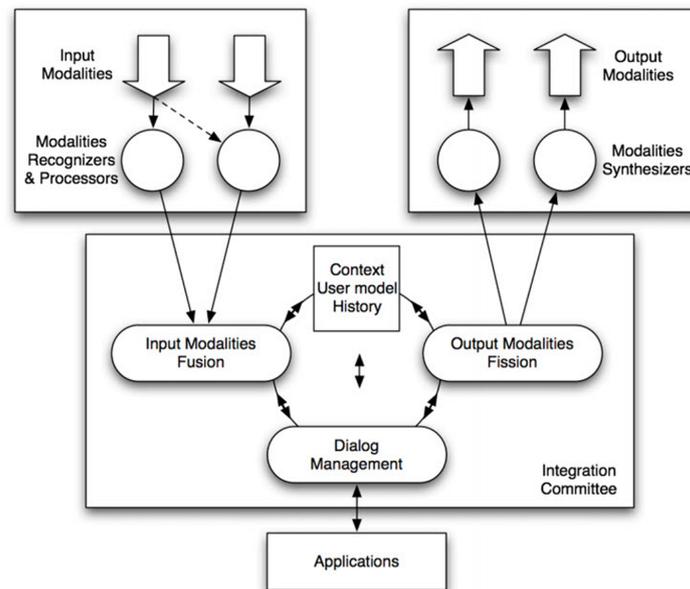


Figure 2.2: Multimodal architecture. Image taken from [32]

Fusion

According to [31, 101, 9], multimodal fusion can be performed at three different levels and use different fusion techniques depending on the moment that the fusion is performed and on the type of information that is going to be fused. Figure 2.3 illustrates the three different levels of fusion.

- ▷ Fusion at the Acquisition Level: Also referred to as *Data Level Fusion*, it comprises the type of fusion that occurs when two or more raw signals are intermixed.
- ▷ Fusion at the Recognition Level: Also referred to as *Feature Level Fusion*, it consists in merging the resulting output from the different input recognisers. According to Dumas et al. [32] this fusion is achieved by using integration mechanisms, such as: statistical integration techniques, hidden Markov models or artificial neural networks. It was highlighted that this type of fusion technique is used for closely coupled modalities like speech and lip movements.
- ▷ Fusion at the Decision Level: Also referred as *Late Fusion*. This type of fusion is the most used within multimodal applications since it allows to

fuse decoupled modalities like for example speech and hand gestures input. The multimodal application calculates local interpretations of the outputs of each input recognisers, then this semantically meaningful information is fused. Three types of architectures are used to implement this type of fusion level, namely: Frame based Fusion [42], Unification based Fusion [51] and Symbolic/statistical fusion [119].

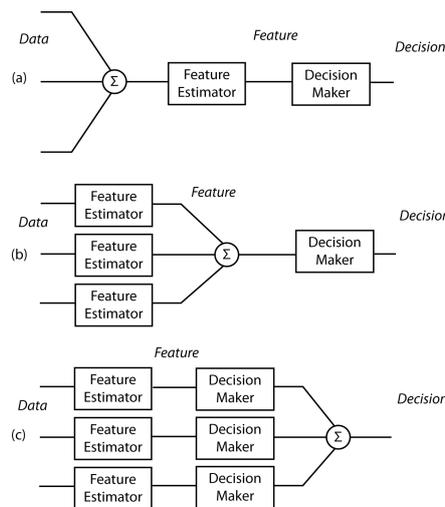


Figure 2.3: Different levels of fusion. Image taken from [101]

Fission

According to Grifoni[43], *multimodal fission* refers to the process of disaggregating outputs through the various available channels in order to provide the user with consistent feedback. Foster[38] describes the fission process in three main steps:

- ▷ **Message construction:** Refers to the process of designing the overall structure of a presentation, specifically selecting and organising the content to be included in the application.
- ▷ **Output channel selection:** Refers to the selection of the most suitable modalities given a set of information. In this phase, it is important to take into account the characteristics of the available output modalities and the information to be presented, as well as the communicative goals of the presenter. A detailed description of these factors can be found in [23].

- ▷ Output coordination: Refers to the construction of a coherent and synchronised result. This step must ensure that the combined output from each media generator correspond to a coherent presentation. The coordination can take the forms of physical layout and temporal coordination referring expressions.

2.2.3 CARE Properties

Besides the components that constitute a multimodal system from an architectural point of view, conceptual models like the CARE model seek to characterise multimodal interaction. This model encompasses a set of properties that deal with modality combination and synchronisation from the user interaction level perspective.

The CARE model was introduced by Nigay et al. [21] and comprises the description of the four types of modalities combination: *complementarity*, *assignment*, *redundancy* and *equivalence*. This model relies on the analysis of the combination of modalities based on two states needed to accomplish a task T , namely the *initial* and *final* state.

Kamel [54] described and illustrated the different properties using as example the following task T : “Fill a text field with the word ‘New York’”. In regards to *complementarity*, two modalities are complementary for the task T if they are used together to reach the final state starting from the initial state. Ideally, modalities are combined to complement the limitations of one modality with the other. Referring to the example scenario, the user might click on the text field with the mouse and then speak the word “New York”. In relation to *assignation*, one can say that a modality is assigned to a task T , if and only if that particular modality allows to fulfil a specific task and there is no other modality that allowed to perform the same action, for instance if the user will only be allowed to speak the sentence “Fill New York” to complete the task T . The property *equivalence* implies that two modalities have the same expressive power, in other words that both modalities allow to reach the final state and perform the task T , only with the limitation that they are not performed at the same time. For example, the user can either click the mouse to select the text field and then select the city “New York” or directly pronounce the phrase “Fill New York”. Finally, the property *redundancy* suggests that two modalities are redundant for the task T if they are equivalent and can be used in parallel to accomplish the task.

2.3 Mobile Interaction

The paradigm shift from desktop to mobile computing started to materialise the vision that Mark Weiser had in 1991 about ubiquitous computing [113].

The extensive research over the past decade on mobile devices hardware and software yielded significant and impressive improvements in the performance, size and cost of these devices. Likewise, from the human computer interaction point of view new research questions have been raised. As explained by Love in [65], mobile HCI is concerned with understanding the type of users and context, their tasks, their capabilities and limitations in order to facilitate the development of usable mobile systems.

2.3.1 Characteristics

The desktop paradigm supposes that users use one single computing device according to their current physical location, for instance one computer at home and another computer at work. On the other hand, the challenge of the mobile computing paradigm is to provide the means that permit users to perform the same task in different physical places using the same device.

The following definitions comprise three important aspects of this paradigm, namely the characteristics of the computing device context, the key enabling technologies and the type of services that can be accessed by the users:

- ▷ “*Mobile computing is the use of computers in a nonstatic environment*”[53]
- ▷ “*Mobile computing refers to an emerging new computing environment incorporating both wireless and wired high-speed networking*”[103]
- ▷ “*Mobile computing is an umbrella term used to describe technologies that enable people to access network services anyplace, anytime, and anywhere*” [50]

These definitions imply that these computing devices must be small enough to be carried around, hence portability and mobility are the key benefits for end users. However, due to these factors, mobile context differs from the desktop and stationary environment in different ways. These differences have been discussed and pointed out by HCI researchers in several works [105, 19, 91]. Thus, to sum up these findings, mobile interaction is characterised by the following constraints and aspects: *limited input and output, multitasking and attention level, context influence and social influence.*

Limited Input and Output

Due to the small size of the device and specifically of the screen display, users have to interact with a limited and new set of input and output technologies. These technologies have been improved over the years to enhance the mobile use experience. For instance, the very first mobile phones used the DTMF keypad, which allowed an easy and fast entry of numeric values but imposed a major difficulty to enter text input. As highlighted by Mauney et al. [69] just for writing the letter “C” a user should press three times the key corresponding to the number 1. Therefore, several techniques based on predictive text have been explored as well as new keyboard technologies like a reduced version of the qwerty keyboard, pen-based input handwriting and virtual keyboards. Although nowadays, virtual keyboards are incorporated in all modern devices, text entry is still very error-prone. According to Henze et al. [44] users suffer from the “*fat finger problem*” since they do not see where they touch and cannot feel the position of the virtual keys and buttons. Other input technologies such as accelerometer-based gestures, the use of tangible interaction or computer vision are explored to expand mobile input techniques.

On the other hand, screen display is still the default output mechanism. Audio and vibrotactile feedback have been explored as alternative output techniques. Mobile display technologies have evolved considerably from their initial presentation. Initial devices had a monochrome CRT display, whereas nowadays devices count with technologies that incorporate AMOLED, LCD or retina displays. These enhancements in display technology helped to notably improve the user output feedback. At the same time, they allowed to explore novel input mechanisms like touch and multi-touch gestures.

Multitasking and Attention Level

Mobile users are mainly doing different types of activities while using their mobile devices including for example driving, walking or working. These activities captures the user attention and mobile tasks always go to a second priority level. As highlighted by Tamminen et al. [105], when an activity is more familiar and working memory is not as taxed, more multitasking can be carried out. Hence, it is important for mobile interaction to minimise the level of attention that the user needs to provide to the screen. According to Chittaro[19], the more attention an interface requires, the more difficult it will be for the mobile user to maintain awareness of the surrounding environment and respond properly to external events which might ultimate, lead to risky situations.

Context Influence

Since one of the challenges of mobile computing is to allow users to use their devices while they are on the go, the surrounding context is a new variable that affects human-computer interaction. Context has been explained multiple times and formally defined by researchers.

Based on the analysis of previous definitions, Abowd et al. [6] defined the term as:

“Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.”

When users are mobile, their surrounding context changes frequently, for example in one single day, a user can be at home, at work, in a street, in the car or on a bus. According to Chittaro et al. [19], the constant change of context has direct implications in user’s perceptual, motor and cognitive levels. Table 2.1 summarises the respective implications.

Level	Implications
Perceptual	* temporally disable the use of some input mechanisms.
Motor	* limits user’s ability to perform fine motor activities. * involuntary movements are produced.
Cognitive	* limits the user’s level of attention to the application

Table 2.1: Context implications in perceptual, motor and cognitive levels. Based on [19]

Social Influence

Even if the cognitive abilities and motor skills allow a user to perform a specific interaction with the mobile device, if the user is in a public place their actions might be conditioned by the task’s level of social acceptability. For instance, as mentioned by Chittaro [19], keeping sound on at a conference is not tolerated, while looking at the device screen is accepted. Other related studies [91, 57] explored the social acceptability of accelerometer based gestures in public places.

According to Williamson et al. [117], these studies seek to evaluate the comfort and personal experience of the performer and the perceived opinions of spectators. For instance, in Rico et al.'s study [91], users were evaluated about their perception of performing a set of motion and body gestures in public locations like home, bus, restaurant and workplace having as audience their partner, friends, colleagues, strangers and family. Results showed that gestures like wrist rotation, foot tapping, shake and screen tapping were considered acceptable to perform in public places. Additionally, familiarity with the audience played a significant role in gesture acceptability. If users are more familiar to the environment and people they are more open to experiment with new interaction techniques.

Therefore, several guidelines have been proposed to address these constraints and distinctive aspects from the mobile setting. Ajob et al. [10] proposed the *Three Layers Design Guideline For Mobile Applications*. The guideline encompasses the three phases of an application's design process, namely analysis, design and testing. The work relied on a thorough analysis of well-known guidelines such as Shneiderman's golden rules of interface design (adjusted for mobile interface design) [102], seven usability guidelines for websites on mobile devices [2], human-centred design-ISO standard 13407 [52] and W3C mobile web best practices¹. Figure 2.5 illustrates the group of guidelines corresponding to each layer.

2.3.2 Mobile Devices

Nowadays users, especially young ones, are very familiar with modern and portable devices. In an user study conducted with 259 participants (average age of 20.6), the familiarity with modern mobile devices was assessed using a questionnaire-based evaluation. The level of familiarity was evaluated using a likert scale ranging from 1 to 5, where five represented *very familiar* and 1 *not familiar at all*. The mean results showed that participants were more familiar with cell phones, laptops, and iPods (M=4.2 – 4.9). Furthermore, participants showed moderate familiarity with tablets and hand-held games such as the portable PlayStation and Nintendo (M=3.2). Finally, it was shown that they were less familiar with PDAs (M=2.9).

To formally categorise all this variety of mobile devices in different groups Schiefer et al. [96] describe a taxonomy of mobile terminals which is depicted in Figure 2.5. Terminals are classified according to the following parameters: size and weight, input modes, output modes, performance, type of usage, communi-

¹<http://www.w3.org/TR/mobile-bp/>

M-G1	ANALYSIS	CONTEXT OF USE (Specify user and organizational requirements)
		<ol style="list-style-type: none"> 1. Identify and document user's tasks 2. Identify and document organizational environment 3. Define the use of the system
M-G2	DESIGN	CONTEXT OF MEDIUM (Produce design solution)
		<ol style="list-style-type: none"> 1. Enable frequent users to use shortcuts 2. Offer informative feedback 3. Consistency 4. Reversal of actions 5. Error prevention and simple error handling 6. Reduce short-term memory load 7. Design for multiple and dynamic contexts 8. Design for small devices 9. Design for speed and recovery 10. Design for "top-down" interaction 11. Allow for personalization 12. Don't repeat the navigation on every page 13. Clearly distinguish selected items
M-G3	TESTING	CONTEXT OF EVALUATION (Evaluate design against user requirements)
		<ol style="list-style-type: none"> 1. Quick approach 2. Usability testing 3. Field studies 4. Predictive evaluation

Figure 2.4: Three layers design guideline for mobile applications. Based on [10]

ation capabilities, type of operating system and expandability. The category *In narrow sense* distinguishes two main groups: *Mobile phones* and *Wireless mobile Computer*.

Mobile phones encompasses the following types of devices: *Simple phones* and *Feature phones*.

Simple phones refer to the classical cellular phone used for voice communication and SMS messages. A *Feature phone* refers to mobile phones with larger display and extended function range than simple phones. However, they do not include extended input modes (only a number keyboard and few additional keys).

On the other hand, *handhelds (PDAs)*, *Mobile Internet Devices* and *Mobile Standard PCs* are categorised under the *Wireless mobile Computer* category. The main distinctive characteristic of *Handhelds* is that they cannot use communication networks for mobile telephony like GSM or UMTS. They have a touch-sensitive display operated with a pen/stylus, text keyboard and navigation keys for input.

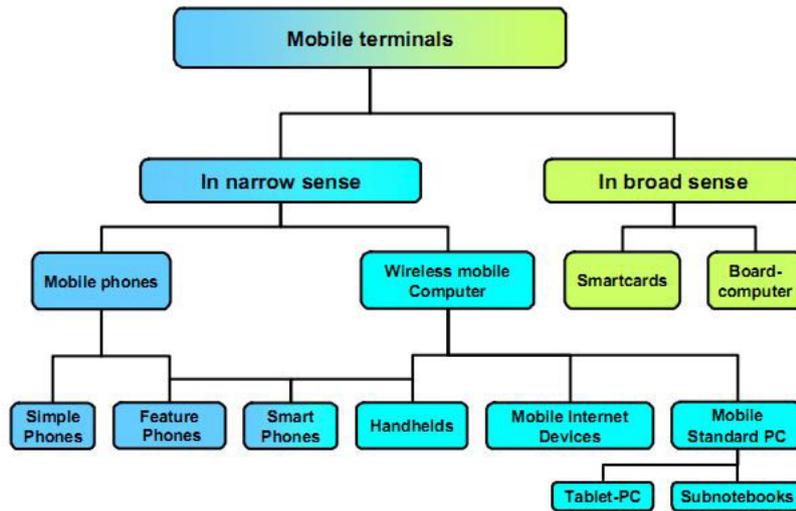


Figure 2.5: Mobile terminals taxonomy. Image taken from [96]

Mobile Internet Devices encompasses devices such as WebTablets or Mobile Thin Clients that are operated through a keyboard. Their main use is web browsing or terminal server sessions. They possess a reduced function range compared to Mobile Standard PCs. In this aspect they are similar to Handhelds. Finally, the *Mobile Standard PC* category refers to devices that use conventional desktop operating systems (Linux, Windows) with compatible software. Laptops, Netbooks and Tablet PCs form part of this category.

Smartphones are categorised between a feature phone and handheld. They are considered as handhelds with the ability to communicate over mobile telephony networks and feature phones that have extended inputs mechanisms provided by a touch-sensitive display or a complete text keyboard. Additionally, Lane et al. [60] highlighted the variety of built-in sensors that current smartphones provide. Figure 2.6 illustrates the most common sensors that come along with new smartphones devices. For example, smartphones like the Google Nexus S or iphone 4 come with built-in sensors such as accelerometers, digital compass, gyroscope, Global Positioning System (GPS), microphone, Near Field Communication (NFC) readers and dual cameras. The authors argued that by combining these sensors in an effective way, new applications across different domains can be researched, for instance in healthcare, environmental monitoring and transportation, thus giving rise to a new area of research called mobile phone sensing.



Figure 2.6: Built-in mobile sensors. Image taken from [60]

2.3.3 Context Awareness

Schilit et al. [97] coined the term context awareness back in 1994, referring to a type of application that changes its behaviour according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time. As stated by Chen et al. [18], context-aware computing is a mobile computing paradigm in which applications can discover and react based on contextual information.

As explained above, Abowd et al. [6] proposed a very broad definition of what context is. On the other hand, Schmidt et al. [98] proposed a context categorization that groups common and similar types of context information in a hierarchical model. The authors categorised context in two main groups, consisting on *human factors* and *physical environment*. Each group was further categorised in *User*, *Task* and *Social Environment* corresponding to *Human Factors*. In turn, *Physical Environment* encompasses factors such as *Conditions* (e.g. noise, light or acceleration), *Infrastructure* and *Location*.

However all applications that gather a user's location information can be categorised as a context-aware application. Abowd et al. [6] argued that it is not mandatory that the application adapts its behaviour based on the context variations. For instance, an application that simply displays the context of the user's

environment like weather or location is not modifying the application's behaviour, yet it is considered as a context-aware application. Based on previous research the authors, pointed out three features that characterise these systems.

- ▷ **Presentation of information and services to a user:** This refers to the ability to detect contextual information and present it to the user, augmenting the user's sensory system.
- ▷ **Automatic execution of a service:** This refers to the ability to execute or modify a service automatically based on the current context.
- ▷ **Tagging of context to information for later retrieval:** This refers to the ability to associate digital data with the user's context. A user can view the data when they are in that associated context.

This paradigm is relevant to the Mobile HCI field because of the mobile nature of mobile users. Since users tend to change their location constantly as well as the persons with whom they interact, their needs and requirements change as well. Dey et al. [28] emphasised that this aspect makes context awareness particularly relevant to mobile computing, since gathering context information makes interaction more efficient by not forcing users to explicitly enter information such as their current location. Thereby, applications can offer a more customised and appropriate service as well as reduce the cognitive workload.

According to Lovett and O'Neill [66], many of the existing mobile context-aware applications focused to gather information regarding the physical location of the user. However, as discussed in the previous section, new built-in sensors allow to infer richer information about the user activity and surrounding environment. Lane et al. [60] explained how these sensors or fusion of sensors data are used in mobile sensing. Among other applications, accelerometers with machine learning techniques are used to classify user activity, such as walking, sitting or running. The compass and gyroscope are used as complementary sensors to provide more information about the position of the user in relation with the device, specifically the direction and orientation. The built-in microphone can be used to determine the average noise level in a room.

Although context awareness is certainly an added value for mobile applications, it also carries potential risks that may affect the application's usability. For example, the users might experiment unexpected device behaviour or "spam" of notifications. Dey et al. [28] proposed a list of design guidelines for mobile context-aware systems. A summary of these guidelines are listed below.

- CA-G1 *Select appropriate level of automation:* If the sensor recognition is known to be very inaccurate for a particular setting, it is advisable not to automate actions in the application.
- CA-G2 *Ensure user control:* The application should provide the user options to alter at any point the actions or information that the system is automatically providing. It is important that he feels having control of the application.
- CA-G3 *Avoid unnecessary interruptions and overload of information:* Due to the lack of attention to the screen that mobile users experiment, it is advisable that the application minimises the number of interruptions and informative messages. In this way, avoiding to compromise the user's attention for unnecessary actions.
- CA-G4 *Appropriate visibility level of system status:* Users should be aware of all the changes in the application context at any time.
- CA-G5 *Personalisation for individual needs:* The system should provide means to modify contextual parameters such as location names or light, noise and temperature limits.
- CA-G6 *Privacy:* Special care should be taken with applications that share sensitive context information like the current location in Google Latitude services. Users should have the possibility to stay anonymous or to only share this information with selected users.

2.4 Adaptive Interfaces

Most of the commercial user interfaces are static in the sense that once they are designed and built they cannot be altered at the runtime. However, due to the heterogeneity of the type of users and their preferences, a lot of research effort has been put to make interfaces more flexible and adjustable to specific user needs or context conditions. What elements of the interface can be adapted, which factors trigger or influence a change in the interface and how the adaptation process occur are key research questions in this field.

2.4.1 Characteristics

User interface adaptation has been the subject of study for more than a decade. According to Vanvelsen [110], personalised systems can alter aspects of their structure or functionality to accommodate the different users's requirements and

their changing needs over time. In a broad sense, user interface adaptation can take place in the form of adaptable or adaptive interfaces. Oppermann et al. [78] explained that the former refers to systems that allow users to explicitly modify some system parameters and adapt their behaviour accordingly. In turn, the latter refers to systems that automatically adapt to external factors based on the systems’s inferences about the user current needs. Figure 2.8 illustrates the whole spectrum of different possible levels of adaptation, having adaptive and adaptable interfaces as reference points.

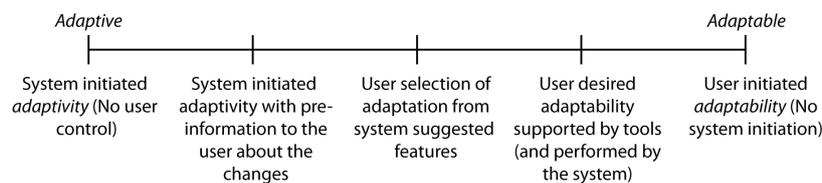


Figure 2.7: Adaptation spectrum. Image taken from [77]

Hence, adaptive interfaces deal with system induced adaptation. Formally, adaptive user interfaces were defined by Rothrock et al. [93] as:

“Systems that adapt their displays and available actions to the user’s current goals and abilities by monitoring user status, the system state and the current situation”

Indistinctly of the type of application, Efstratiou [34] highlighted that three main conceptual components characterise an adaptive system, namely the *monitoring entity*, *adaptation policy* and *adaptive mechanism*. These components are analogous to Opperman’s afferential, inferential, efferential core component of an adaptive system [76].

Monitoring Entity

Adaptive systems can gather data from multiple sources. Hence, this component is responsible of permanently observing specific contextual features that might indicate to the system that the adaptation process must start.

Adaptation Policy

This component is in charge of evaluating and analysing the gathered data from the monitoring entity. It decides in which way the system should modify its be-

behaviour evaluating a set of predefined rules or using heuristic algorithms. Opperman [76] refers to it as the switchbox of an adaptive system.

Adaptive Mechanism

This component deals with the system modifications when an adaptation call is triggered. The adaptive mechanism is in charge to perform the corresponding modification in the presentation or functionality of the system. This component is tightly coupled with the semantics of the application. Malinowski et al. [67] highlighted that the possible adaptive mechanisms are *enabling*, *switching*, *reconfiguring* and *editing*. *Enabling* refers to the activation/deactivation of system components such as turning on/off audio input. *Switching* refers to an interface modification based on the selection of one of the multiple feature values within the user interface, for example changing the background colour from white to gray. *Reconfiguration* refers to a modification in the organisation of the elements in the interface and *editing* encompasses a modification without any restrictions.

According to Bezold et al. [12], the goal of automatic adaptation is to improve the overall usability and the user satisfaction of the application. Based on findings from previous work, Wesson et al. [115] and Lavie et al. [61] summarised the main benefits of these type of interfaces. In a broad sense, these systems can improve task accuracy and efficiency. Likewise, they help to reduce learnability and minimise the need of users to request help. Additionally, they are an alternative solution for problems such as information overload and filtering, learning to use complex systems and automated task completion.

These benefits are achieved only when specific aspects are taken into consideration during the design and development process. Gajos et al. [39] highlighted the following factors that influence user acceptance of adaptive interfaces, namely the *predictive accuracy of an adaptive interface* and the *frequency of the adaptation*.

The predictive accuracy of the adaptive interface refers to the correctness of the results provided by the system. If a change in the interface is expected and does not occur, users start to feel confused and the level of predictability goes down too. *The frequency of the adaptation* refers to how fast and often a change in the interface is perceived by the user. Slow-paced adaptations have much better user acceptability than fast paced adaptations. Furthermore, their results showed that the frequency of the interaction with the interface and the level of cognitive load demanded by the task affects the aspects that users consider important in the interface. For instance, if a task is commonly used by the user and also encompasses a cumbersome process, the user perceives an added value if the system helps him to perform the task in a quicker or easier manner.

Furthermore, Rothrock et al. [93] presented guidelines that support the process of adaptive interface design. It comprises three important points:

- A-G1 *Identify variables that call for adaptation:* The authors specify nine variables that commonly influence adaptation and are classified based on the physical origin of the input, namely *user*, *current situation* and *system variables*. Examples of variables for the *user* category are user knowledge, performance or abilities. In turn, examples of the *situation* variables category are noise, weather, location in space and location of targets. Finally, an example of the *system* category variables is any change in the state of the system.
- A-G2 *Determine modifications to the interface:* The designer should determine how and when the content of the interface should adapt to the calling variables. In this section four categories should be taken into account, namely the *content to be adapted*, the *structure of the human-system dialogue* or navigation (commonly used in hypertext context), *task allocation* in terms of automation levels and the *moment and speed of the adaptation*.
- A-G3 *Select the inference mechanism:* The designer should select an appropriate inference mechanism, for example they can choose to use a rule-based mechanism, predicate logic or a machine learning-based classifier approach. Indistinctly of the selected approach, the mechanism should be able to fulfil the two functionalities of identifying instances that call for adaptation and deciding on the appropriate modifications to display.

2.4.2 Conceptual Models and Frameworks

Different frameworks and models have been presented to describe adaptation design and run time phases without taking into account specific implementation requirements.

The conceptualisation of the adaptation process has been addressed by several authors, for example Malinowski et al. [67] presented a complete taxonomy of user interface adaptation. The authors described a classification of the main concepts in the field such as the stages and agents involved in the process, types and levels of adaptation, scope, methods, architecture and models. They describe four distinguished stages that describe the adaptation process, namely *initiation*, *proposal*, *decision* and *execution*. These stages can be performed either by the user or the system. Figure 2.8 illustrates an example of a possible combination of the

responsible agent for each stage. A similar approach has been proposed by Lopez-Jaquero et al. [64] under the research of the Isatine Framework. This framework, besides describing the different stages of the adaptation process, includes a stage which specifies how the adaptation process can be evaluated to meet the adaptation goals.

	System	User	
Initiative	●		system initiates adaptation
Proposal	●		system proposes some change/alternatives
Decision		●	user decides upon action to be taken
Execution	●		system executes user's choice

Figure 2.8: Adaptation process: agents and stages. Image taken from [67]

However, as stated by Bezold et al. [12], some stages as for example, *the initiative for adaptation* or *decision* are redundant to describe a fully adaptive system process. Therefore, Paramythis and Weibelzahl [84] presented a framework to describe specifically the system induced adaptation process. A description of each stage is listed below and illustrated in Figure 2.9

- ▷ Monitoring the user-system interaction and collecting input data: The data that the system collects in this stage comes from user events and from the data gathered from different sensors. However, this information does not carry any semantic meaning for the application.
- ▷ Assessing or interpreting the input data: In this stage, the collected data should be mapped to meaningful information for the application. For instance, if the GPS sensor indicates that the number of satellites sensed to identify a user location is less than two, this numeric value might indicate that the user is situated in an indoors location. However, this value can have a totally different meaning in the context of another system.
- ▷ Modelling the current state of the world: This refers to the design and population of dynamic models that will contain up-to-date information of relevant entities related to the user, context and interaction history.
- ▷ Deciding about the adaptation: Based on the up-to-date information provided by the models, the system decides upon the necessity of an adaptation.

- ▷ Executing the adaptation: This stage refers to the transformation of high-level adaptation decisions to a specific change in the interface perceived by the user.
- ▷ Evaluation: similar as in the Isatine framework, in this stage the overall adaptation process has to be evaluated. Designers are encouraged to list the reasons that motivate the use of adaptation in the interface. Then, at the end of the design process, evaluate if these goals were satisfied.

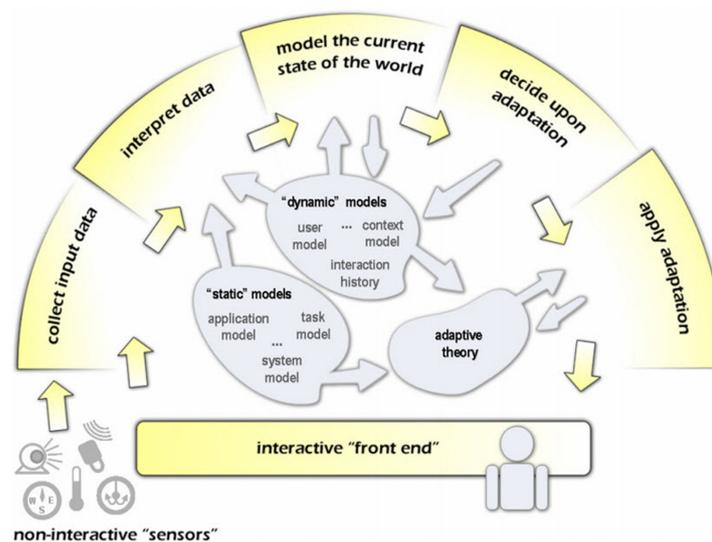


Figure 2.9: Adaptation decomposition model. Image taken from [84]

Finally, important concepts were introduced by Calvary et al. [16] within the CAMELEON framework research, specifically the concepts of plasticity and multi-targeting. Plasticity refers to the capability of an interface to preserve the usability while adapting its interface to multiple targets. Multitargeting encompasses the different technical aspects of adaptation to multiple contexts. Contexts denote the context of use of an interactive system described in terms of three models including user, platform and environment. The user model contains information about the application's current user, for example user preferences or limitations such as disabilities. The platform model describes physical characteristics of the device where the system is running on, for example the size of the screen or processor speed. Finally, the environment model contains information about social and physical attributes of the environment where the interaction is taking place. This model encompasses three categories: *Physical Conditions* (e.g. level of light,

pressure, temperature, noise level and time), *Location* (e.g. absolute and relative positions and co-location), *Social Conditions* (e.g. stress, social interaction, group dynamics or collaborative tasks) and *Work Organization* (e.g. structure or a user's role).

2.4.3 Adaptivity in Mobile and Multimodal Interfaces

A lot of the research work related to system- induced adaptation has been done in desktop environments, in stand-alone as well as in web applications. However, in the past few years, more interest has been put in system-induced adaptation in the domain of mobile and multimodal interfaces.

Due to the steady growth of mobile computing, system-induced adaptation has been researched in this setting as well. It has been highlighted in [72, 24, 18] that the ability to adapt to change of context is critical to both mobile and context-aware applications.

Among mobile applications, the importance of automatic adaptation has a big relevance because reducing mobile users cognitive load is a paramount aspect in this setting. Thus, this type of interfaces is an alternative to deal with this constraint. Mostly, mobile adaptive interfaces applications adapt their behaviour based on interaction context variations (user, environment, device). For instance, Apointer, a mobile tourist guide [45] allows to search points of interest such as restaurants or accommodations relying on adaptation techniques. Thereby, the displayed map information as well as the zoom functionality rely on the current location provided by the GPS sensor data. Additionally, user actions are stored in a history queue and used to reorganise the interface components based on frequency and recency of use. Similarly, other domains like education [36] and healthcare [68] have explored the use of adaptive interfaces in mobile settings.

Likewise, in several works related to the multimodal interfaces field, the importance of the automatic adaptation of input and output modalities has been highlighted. From an architectural point of view Lalanne [59] encouraged to further study the dynamic adaptation of fusion engines based on the ongoing dialogue and environmental context. Oviatt [81] argued that future multimodal interfaces, especially mobile ones, will require active adaptation to the user, task and environment. Furthermore, Chittaro [19] claimed that context awareness within multimodal applications should be exploited in order to reduce attention requirements and cognitive workload.

He highlighted that adaptation should deal with three aspects: the information

the device should present, the best modality or combination of modalities based on the task and context and finally the functions that could be useful or wanted by the user in his current situation.

In this field, initial studies have been driven by Duarte et al. [30], who described a conceptual framework called FAME for designing adaptive multimodal applications. FAME's adaptation is based on context changes and relies on the Cameleon framework models: *user*, *platform* and *environment* and an extra model called *interaction model*. Additionally, in this work a set of guidelines and the concept of the *Behavioural Matrix* are introduced. The behavioural matrix aims to support the designer during the process adaptation rules definition. The "Desktop Multimodal Rich Book Player (DTB Player)" application was presented to illustrate the capabilities of the framework. The application allowed to adapt the available output modalities. The available output modalities were *visual for presenting text and images* and *audio* for playback and speech synthesis. For instance, for the presentation of the miscellaneous components such as annotations, if the content was displayed using visual output then the main content narration continued. In turn, if the presentation of the content used audio output the main context narration paused.

Chapter 3

An Investigation of Mobile Multimodal Adaptation

In the previous chapter, multimodal, mobile and adaptive interfaces were reviewed in detail by highlighting the features and characteristics of their interaction styles. The mobility of mobile users makes multimodal and adaptive interfaces a good complement to enhance mobile interaction. Recent research has explored the use of multimodal interfaces in the mobile context, analysing the challenges and benefits that the combination of these two interaction paradigms imposes to users and developers.

This chapter begins by giving a short introduction about the motivation and scope of this study. Afterwards, the description of the related work within the scope of the study is described as well as the parameters that are used to classify the selected research work. Finally, recapitulative tables along with an analysis section are provided.

3.1 Objectives and Scope of the Study

Initial studies in the field of *Multimodal Mobile Interfaces* were headed by Oviatt in [82, 80]. Further research work on mobile multimodal interfaces has focussed on defining guidelines and conceptual frameworks to ease the design and development process of mobile multimodal interfaces [22, 19, 58]. Additionally, frameworks that allowed to evaluate such interfaces by measuring statistics about users's modality usage and also evaluating how users react under distracting and stressful conditions were addressed by different authors [100, 8].

A new and common research direction for mobile as well as for multimodal interfaces is the system-induced adaptation. Although the importance of auto-

matic adaptation within multimodal mobile applications has been shown in the former chapter, the field has not yet been fully explored. Automatic adaptation in this domain has been explored mostly by adapting the output modalities either to users [55, 20] or to context [88, 17]. The field of input adaptation has been neglected until now, probably due to hardware limitations related to mobile input mechanisms. However, current devices offer a broader range of input modes that enables and promises more active work in this field.

Therefore, this study seeks to make an exhaustive analysis of multimodal mobile input channels with a special focus on adaptation triggered or influenced by environmental factors. The following main aspects are addressed:

- ▷ **The modalities or combination of modalities that are used in the multimodal mobile setting.** It has been determined that modern mobile devices allow users to interact using new and different input mechanisms. Therefore, it is important to investigate how input modalities are used and combined in the mobile setting. By having an overall picture of the available and possible input modes, it will be possible to discover promising areas of research. At the same time, this analysis provides a set of modalities that could be used by an adaptation mechanism.
- ▷ **The influence of environmental factors in the selection of the optimal input modality.** In the mobile interaction literature section, it was observed that this type of interaction is constrained by factors like user limited attention to the device as well as from the influence of contextual factors. The focus of this analysis concentrates on investigating how mobile multimodal systems are addressing environmental influence and which modality is preferred under specific environment properties. The outcome of this analysis provides us with an insight about which modality should be used or avoided in a particular contextual situation. This information could be useful as a conceptual basis to automatise the selection of optimal input modalities in an adaptation process.
- ▷ **Mobile multimodal automatic adaptation.** The main focus of this analysis is to review the system-induced adaptation of input modalities channels, specifically to analyse the following two points: to what exactly these systems adapt and which are their monitoring entity, adaptation policies and mechanisms. Based on these findings, a concise summary is presented describing the different ways that mobile multimodal input adaptation can take place.

The scope of the study is clearly distinguished in Figure 3.1. The study focus of interest finds itself in the intersection between three main areas: *mobile multimodal input*, *system-induced adaptation* and *environment properties*. Thus, the research work included in this study will be constrained to investigations relevant within the shaded area marked with the number 1. However, the study additionally includes research works related to mobile multimodal input adaptation induced by the user and influenced by environmental changes. Research work under this section, highlighted with the number 2, deals with modality selection and context management and provides a conceptual basis for the automatic adaptation research work. Therefore, special attention has been put to select research work that, even though they do not present automatic adaptation, take into account context influence as part of their study.

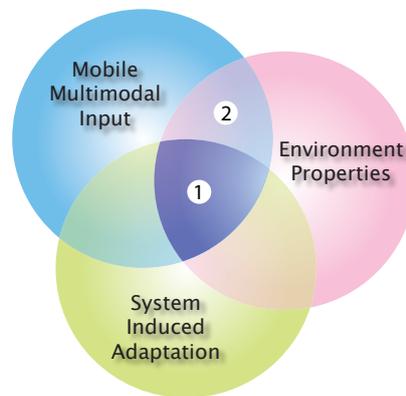


Figure 3.1: Scope of the study

The ultimate goal of this study is to draw conclusions based on the three aforementioned partial analyses. This information allows to establish a set of core features that facilitate the process of designing and developing an adaptive context-aware multimodal mobile application. These features are the basis to design and develop the proof of concept application described in Chapter 4.

3.2 Study Parameters

The research work that met the selection criteria was classified using parameters that describe main features related to multimodal interaction, mobile interaction, context awareness and the field of user interface adaptation. Specifically, the parameters *modalities*, *interaction techniques*, *interaction sensors*, *output influence*

and *CARE properties* are related to multimodal interaction. The parameters *device* and *environmental conditions* are related to mobile and context awareness concepts and finally the parameter *adaptation* presents information relevant to this field. This categorisation is the basis to perform a systematic analysis of the selected research papers. A detailed description of the parameters is listed below.

- ▷ **Modalities** describes which modalities are proposed by the described system. *2D gestures* describe gestures or interactions that are performed using a finger on a touch screen. *Pen gestures* refer to gestures and interactions that are executed with a small pen whereas *Motion gestures* represent gestures performed in free space with the phone in the hand and which are recognised by accelerometers. *Extra gestures* are linked to some tangible interaction which can for example be based on QR tags or RFID-tagged objects. *Speech* designates some speech recognition software and last but not least *Indirect manipulation* refers to the use of the keypad, special keys and keyboard of the device.
- ▷ **Interaction Techniques** designates the type of interaction which was used for each modality. For example, in the case of speech, sometimes predefined voice commands are used, whereas other systems support natural dialogue interaction.
- ▷ **Interaction Sensors** describes which hardware sensors are used to recognise the specific modalities. Accelerometers or digital compasses are examples of sensors that are used to determine the orientation of a smartphone and, in turn, support the recognition of motion gestures.
- ▷ **Devices** specifies on which class of device and on which operation system (if this information was available) the system was running. The used taxonomy is presented in [96].
- ▷ **Output Influence** lists a system's output modalities. It also describes whether the selection of the input modality had any influence on the selection of the output modality.
- ▷ **CARE Properties** reports which temporal combinations described by the CARE model were taken into account at the fusion level.
- ▷ **Environmental Conditions** lists the context information which was used by a system. These are based on the properties put forward in the Cameleon framework [16] presented in the background section.

- ▷ **Types of applications** details the targeted audience or domain of application.
- ▷ **F/M/A** specifies if the work presented in an article is a framework, middleware or application.

Extra parameters are taken into account for the study of the research work related to system induced input adaptation. The following parameters attempt to characterise in detail how the adaptation process was performed.

- ▷ **ME** refers to the *Monitoring Entity* component, specifically to the sensor that captures information that will be used to decide if an adaptation should occur or not.
- ▷ **AP** refers to the *Adaptation Policy* component and comprises the set of rules or heuristics that permits to evaluate if a change in the system should be triggered.
- ▷ **AM** refers to the *Adaptation Mechanism* component. If the rules or heuristics result in a true value, information about how the application performs the adaptation process is presented.

3.3 Articles Included in the Study

Articles listed in this section describe prominent research work from the past 10 years related to the field of mobile multimodal input adaptation influenced by environmental factors. The first section presents an overview of *user-induced adaptation* and the second section is devoted to *system-induced adaptation*. Each section first describes existing frameworks and methods that facilitate the design and development process of mobile multimodal applications. Subsequently, research work that is devoted to explore different applications domains is presented.

3.3.1 User-Induced Adaptation

The flexible nature of multimodal systems makes these systems adaptable by default, in other words these systems can alter the current input mode of the application according to explicit user input events. This section outlines the state of the art in the *mobile multimodal input* field with a focus in research work where environmental properties are taken into account as parameters that influence the modality selection. Thus, it entails the area delimited with number 2 in Figure 3.1. Table 3.1 depicts a summary and classification of the main features from the articles described in this section.

Name	Modalities	Interaction Techniques	Interaction Sensors	Device	Output Influence	CARE Properties	Environment Conditions	F/M A	Application Domain
[112] Wasinger et al. 2005	* Speech * Extra Gestures (see [112]) * Pen Gestures * 2D Gestures	* Voice Commands * Pick up * Put back * Handwriting * Pointing * Single Tap	* Microphone * RFID Reader * Stylus * Touch Display	Wireless Mobile Computer Handheld (PDA)	No * Graphical Output	* Complementarity Voice Commands + Single Tap Voice Commands + Pointing Voice Commands + Pick up (Compare two products information) * Equivalence Single Tap Pointing Pick up (Select a product from a list of items) * Redundancy Voice Commands + Pointing (Ask for the characteristics of an item)	* Physical Conditions: Noise Level: medium * Social Conditions: Crowded environment * Location: Public places (electronics store)	A	Services (Shopping Application)
[104] Sonntag et al. 2007	* Speech * Pen Gestures	* Natural Dialogue * Pointing	* Microphone * Touch Display	Wireless Mobile Computer Handheld (PDA)	No * Graphical Output * Audio Output	* Complementarity Natural Dialogue + Pointing (Ask for information about a player) * Assignment Natural Dialogue (Ask general & deitic questions)	* Physical Conditions: Time: Current Date * Location: GPS absolute location	A	Services (SmartWeb Q/A FIFA World cup 2006 guide)
[62] Lemmela et al. 2008	* Speech * Motion Gestures * 2D Gestures	* Voice Commands * Tilt up, down, left and right * Finger Stroke * Single Tap	* Microphone * Ext. Accelerometer * Touch Display	Wireless Mobile Computer Driving Mobile Standard PC Walking Mobile Internet Device	Yes * Graphical Output * Audio Output * Vibra Feedback	* Assignment Voice Commands (Driving) * Equivalence Finger strokes Tilt up, down, left, right (Browse messages) Voice Commands Single Tap (Select "Reply Message" option)	* Driving * Social Conditions: Stress (Avoid cars) * Location: Car Walking * Social Conditions: Social Interaction * Location: Office areas * Physical Conditions: Noise Level: medium	F	Communication (SMS application)
[29] Doyle et al. 2008	* Speech * Pen Gestures	* Voice Command * Dictation * Pointing * Handwriting * Dragging	* Microphone * Stylus * Touch Display	Wireless Mobile Computer Mobile Standard PC (Tablet PC)	No * Graphical Output	* Equivalence Dictation Handwriting Pointing (Correct errors during dictation of annotations) * Complementarity Voice Command + Pointing (query map) Voice Dictation + Pointing (Create annotations in the map) * Assignment Voice Commands (Navigate and query spatial features) Dragging (Zoom in)	* Physical Conditions: Noise Level: low, high * Location: Indoors Canteen Outdoors	A	Map & GIS (Compass system tourist guide)
[89] Reis et al. 2008	* Speech * Pen Gestures * 2D Gestures * Indirect Manipulation	* Voice commands * Pointing * Symbol drawing * Single tap * Symbol drawing * Keypad navigation	* Microphone * Stylus * Touch Screen * Keypad	Wireless Mobile Computer Handheld (PDA) Mobile Standard PC (Tablet PC)	No * Graphical Output * Audio Output	* Equivalence Voice Commands Pointing Single Tap Keypad Navigation Symbol drawing (Select menu options: next, previous, up, play, down, play, record)	* Location: Home Parc Subway Car (Driving) * Social Conditions: Crowded environment * Physical Conditions: Noise Level: low, medium	A	Services (Form- Filling Application)
[26] De Sa et al. 2009	* Speech * Pen Gestures * 2D Gestures * Indirect Manipulation	* Voice commands * Text Selection * Pointing * Single Tap * Keypad Navigation	* Microphone * Stylus * Touch Screen * Navigation	Handheld/ Smartphone (Mixed Fidelity Prototype)	Yes * Audio Output * Graphical Output	* Equivalence Single Tap Keypad Navigation Voice Commands (Select menu options) * Complementarity Text selection + Voice Commands (Create annotations in the book)	* Location: Living Room Elevator Street Subway * Physical Conditions: Noise Level: low, high Light: well-lighted * Social Conditions: Crowded environment	F	Entertainment (Mobile Digital book)
[86] Ramsay et al. 2010	* Motion Gestures * Indirect Manipulation	* Tilt up, down, left and right * Rotate left and right * Keypad Navigation	* Ext. 3 axis accelerometer * Keypad	Smartphone (Symbian) + Shake device	No * Graphical output	* Equivalence: Key up/down, Tilt up/down (Scroll map) * Assignment: Rotate left and right (Explore a location in the map - open/close POIs)	* Physical conditions: Light : well-lighted Noise Level: medium * Social Conditions: Collaborative tasks Stress (Answer calls and walk) * Location: Parc	A	Map & GIS (Tourist Guide)
[92] Ronkainen et al. 2010	* 2D Gestures * Indirect Manipulation	* Double tap gesture (back of the device) * Typing	* Accelerometer * Keyboard	Feature Phone (Symbian Nokia 5500 Sports)	Yes * Vibra Feedback * Audio Output	* Assignment: Double Tap (Silence phone, trigger speech synthesizer to read a message) Typing (Write message)	* Location Street (cycling) Car (driving) * Physical Conditions Noise Level: high Weather: winter	F	Communication (SMS / PhoneBook)

Table 3.1: User-induced adaptation in mobile multimodal systems

Design and Development: Tools and Methods

Conceptual frameworks and methodologies to design adaptable mobile multimodal applications rely on a systematic analysis of context variables and scenarios. In this way, it is ensured that users can switch between suitable modalities when environment properties change.

For instance, Lemmela et al. [62] proposed an iterative method to design multimodal applications in mobile contexts. The interesting contribution of this approach is that they identified which modalities and combinations of modalities best suit different mobile situation requirements based on the user's sensory channels load. The process comprises five steps: The first step of *identifying iteration limitations of mobile situations* relies heavily on context analysis. The authors first suggest to identify common mobile contexts like walking, driving a car, being in a gym, shopping groceries, having lunch in a cafeteria or travelling by bus. Then, for each context the user's aural, visual, physical and cognitive required load should be analysed and depicted using a four point likert scale. These user limitations are caused by different factors, for instance *aural load* is increased by the traffic noise or surrounding people noise, *visual load* is caused by the amount of natural light or traffic lights. Thus, a user's visual load is higher in a driving car context than in a walking context. In turn, *aural load* is higher in a walking context than in a driving environment. With this information, designers have an insight about which modality might be suitable for each scenario. The second step, *identifying and selecting suitable interaction concepts and creating a general design*, depends on the information from the context analysis. Using this information, a set of use cases and scenarios are depicted illustrating basic tasks and contexts of use. To support this process, the authors provided a summary of the characteristics, limitations and strengths of different output modalities. Then, in step 3, *creating modality-specific designs*, using the modalities characteristics collected in the previous steps, specific design decisions should be taken for the creation of the user interface. For instance, in the car context, a specific system-directed speech interface was designed whereas in a pedestrian environment a mixed strategy was used. Last steps comprise the rapid prototyping and evaluation, respectively. For test purposes, an SMS application was developed and evaluated in a car as well as in a pedestrian context. Speech input was assigned as the default interaction technique for the car environment whereas 2D gestures (finger strokes) and motion gestures (tilt gestures) were used in the walking environment. In both scenarios, users had to write and read SMS messages while doing other activities. It was pointed out that users prefer to use the speech input modality while being in the car context.

On the other hand, they preferred 2D gestures and motion gestures during walking because these modalities demand less visual attention.

De Sá et al. [26] describe a set of techniques and tools to support designers in the creation of mobile multimodal applications. The one related to context and environmental influence is a conceptual framework called *Scenario Generation and Context Selection Framework* [27]. The framework aims to facilitate the process of selection and generation of scenarios in a mobile setting. It relies on the analysis of a set of variables that might affect user interaction. According to the authors, by identifying and analysing these scenarios, alternative or complementary modalities can be introduced to overcome temporal limitations. The framework is built upon three main pillars, namely: *contextual scenarios*, which refer to the scenarios composed by instantiating *scenario variables* in a particular mobile setting. *scenario variables* encompass specific aspects that compose each contextual scenario. These variables are classified in five main groups: *locations and settings*, *movement and posture*, *workloads*, *distractions and activities*, *devices and usages* and *users and persona*. Particularly, the category *location and settings* comprises environmental factors like lightning, noise, weather conditions as well as the social environment in which the user is located. Finally, *scenario transitions* refer to the changes from one contextual scenario to another. For instance, when the user moves from the bedroom to the kitchen and starts to interact with fingers instead of the stylus. The approach was evaluated during the design phase of the rich multimodal mobile Digital TalkingBook player. This application supported the use of speech, 2D and pen gestures as well as indirect manipulation. Different contextual scenarios were presented during the design process. For instance, in two of the contextual scenarios, the authors describe Jane, a visually impaired person, changing the input modality of the application depending whether she was at home or in an elevator. The first scenario was composed using the following scenario variables: *persona* (Jane), *location* (Living Room), *position* (seated), *usage* (stylus input) and *environment* (silent and lighted). A scenario transition specified a change in the input modality from dual-handed with stylus to single handed with finger based interaction. The final contextual scenario specified the following information: *persona* (Jane), *location* (Elevator), *Position* (Standing), *Usage* (Finger input) and *environment* (silent, well lit). According to the authors, the scenarios illustrate how the changes in context variables influence the usage of multiple modalities.

Finally, Ronkainen et al. [92] proposed a conceptual framework called *Environment Analysis Framework* to perform a systematic environment analysis. The framework was built based on the analysis of previous work regarding mobile usage and context influence. The authors claimed that the output of the framework

can be used to guide the design of adaptive and/or multimodal user interfaces or devices optimised for certain usage environments. The framework relies on four concepts: *environment variables*, *effect*, *resource variables* and *design idea*. *Environment Variables* refers to an aspect of the environment that can be measured or logged. For instance, the ambient light luminance during a sunny day is 60.000 lux. *Effect* refers to the effect that the variable has on the user/device interaction. For instance, a high luminosity level leads to a bad visibility of the display. *Resource variables* comprise the capability of the user or a mobile device on which the environment variable has an effect. For example, the capability to use both hands, one hand or no hand at all. Finally, *design idea* refers to initial ideas for overcoming the environment effects restricting device usage, for example replace or complement visual feedback with auditory output.

The environment variables that the authors defined are the following: *level and spectrum of background noise*, *babble noises*, *sporadic noises*, *social context*, *cognitive load*, *something blocking the ears*, *typical storage place for device* and the *need to listen to the environment*. Additionally, a list of the environment variables used in the analysis of input and output modality was provided.

Thus, based on these concepts, the framework can be summarised in the following steps. The first step is to select the user's and device's resources to be analysed. For each resource, related resource variables have to be defined. For instance, one resource variable associated with the resource speech input is the *ability to speak in the environment*. Then, mappings between resource variables and environment variables must be defined. For instance, for the previous resource variable a corresponding environment variable could be the *level and spectrum of background noise*. The third step comprises an environment analysis of the mappings defined in step two. For example, the effect of the environment variable *level and spectrum of background noise* is that it affects the overall speech recognition. Thus, the *ability to speak in the environment* is difficult. Finally, the "resource space" should be defined. This space quantifies the demands that the environment places on the user resources for a given task. In this context, user resources refer to the capability of using a certain modality as well as the level of cognitive load and social interaction. For example, for the task *walking in a busy street*, the resource *how well the user can use the speech input* would be assigned the value of difficult because high levels of noise are expected. It is important to note that the dimension of the resource space is equal to the number of resources selected for the analysis..

The framework was populated and evaluated having as main studied environment *bicycling in Beijing*. For this environment, the analysed input modalities were 2D gestures and speech. Finally, the tasks that were evaluated for the selected environment were common mobile tasks like: answering an incoming call

or reading a received text message. The environment analysis confirmed the high level of difficulty of manual input while cycling, hence the double tap gesture was proposed and evaluated among users.

Applications

Some works also addressed how the influence of context, specifically environment properties, affects users interaction and preferred modalities in different applications domains.

For instance, the map-based applications domain has been analysed by several authors. For example, Doyle et al. [29] conducted a review and analysis of existent map-based multimodal systems. They further proposed and evaluated a new multimodal mobile geographic information system (GIS) called Compass. Parameters such as effectiveness and efficiency over unimodal interaction were evaluated. Additionally, to evaluate the effect of changing environmental conditions, user evaluations were conducted in outdoor and indoor environments as well as in quiet and noisy environments. Users were able to interact with the system using speech and pen gestures modalities in a complementary and equivalent manner. It was highlighted that the increment of recognition errors using a multimodal mobile approach in noisy environments decreased the overall interaction speed in comparison with quiet environments. Thus, the use of speech under such environmental conditions is not advisable. To deal with the complexity of working with digital maps on small screen displays, Ramsay et al. [86] proposed the use of motion gestures like tilting an external device backwards and forwards to navigate within a map. The study focussed on evaluating user preferences and perception about the new input interaction techniques in comparison with traditional keypad navigation. To conduct the user evaluation, a tourist guide application was developed and tested in the field. The application allowed users to find and explore points of interest on the map while they were walking through a park. During the tests, the input modalities were used in an equivalent way by the users. The results showed that basic map navigation movements like scrolling left, right, up and down worked well with the keypad and the external shakeable device. Since the evaluations were performed in the wild, users commented that they perceived an added value of using gestures in outdoor environments.

Other typical mobile outdoor activities like shopping, accessing different kinds of web services and forms-filling tasks were also explored. For instance, the goal of Wasinger et al.'s work [112] was to explore the use of tangible interaction as a complementary input modality for speech, 2D gestures on touch displays and pen gestures. They developed a proof of concept application to measure the in-

tuitiveness as well as to evaluate user's preferred combination of modalities in a shopping context. Users were also asked about the influence of the change of environment for the selection of their preferred input modality. Their results show that in public environments users feel comfortable using 2D gestures, extra gestures (tangible interaction) as well as pen-based interaction. In turn, in private environments users feel comfortable using all modalities indistinctly.

On the other hand, under the research of the SmartWeb project, Sonntag et al. [104] investigated the use of natural language and multimodality as an interface to intuitively retrieve different types of information from web services. An important aspect of the system is its context-aware processing strategy. All recognised user actions are processed with respect to their situational and discourse context. An application scenario of the system was a question-answering dialogue system that allowed users to query information from World Cup players and games stored in an ontological knowledge base. Related information like the current weather forecast was retrieved using web services. Speech was assigned as the predominant input modality, whereas speech and pen-based gestures were used in a complementary fashion to select information between different available options. Context information was taken into account in the speech utterances by supporting time and location deictic expressions such as "*How is the weather going to be like tomorrow?*" or "*How do I get to Berlin from here?*". The meaning of "*tomorrow*" and "*here*" was interpreted according to the values obtained from the current date and a GPS sensor.

Finally, Reis et al. [89] investigated the preferred user modality under different mobile environments. The authors presented a mobile multimodal application that allowed the users to answer questionnaires and fill information. The test was conducted in four environments including *home*, *parc*, *subway* and *car*. The evaluated input modalities were 2D and pen gestures, speech and keyboard based input. Their results showed that in quiet environments without the presence of strangers and other disturbing factors, users were eager to experiment with new modalities.

3.3.2 System-Induced Adaptation

Although most of the research on adaptation has shown input adaptation initiated by the user, some research work addresses the automatic activation or deactivation of input modalities in a given context. According to specific contextual variations, the system infers how and when different modalities should be switched on and off. This section entails the area delimited with number 1 in Figure 3.1. Table 3.2 presents a summary of the main features from the reviewed articles.

Name	Modalities	Interaction Techniques	Interaction Sensors	Device	Output Influence	CARE Properties	Environmental Conditions	F/M/A	Adaptation	Application Domain
[14] Böhler et al. 2002	*Speech *Pen Gestures	*Natural Dialogue *Pointing	*Microphone *Stylus	Wireless M . Computer (Ipaq Handled) + Mobile Standard PC	Yes *Graphical Output *Audio Output	*Complementarity Natural Dialogue + Pointing (Ask for information about a place in the map) *Equivalence Default Mode: Natural Dialogue Pointing *Assignment Speech Mode: Natural Dialogue Silent / Listener Mode: Pointing	<u>Driving</u> *Social Conditions: Stress *Location: Car <u>Walking</u> *Physical Conditions: Noise Level : high *Social Conditions: Social Interaction Stress *Location: Street	F	*ME: Driving speed State of brakes Location Noise level *AP: Rule-based *AM: Switch the default input mode	Map & GIS (Smartkom System)
[85] Porta et al. 2009	*Speech *2D Gestures *Motion Gestures	*Natural Dialogue *Single Tap *Shake	*Microphone *Touch Display * Accelerometer	Smartphone (iOS)	No *Audio Output *Graphical Output	*Complementarity Single Tap + Natural Dialogue (Ask for a specific service) *Assignment Natural Dialogue (Ask for information about services) Shaking (Undo task)	*Location: Relative Location (Device-user mouth) Outdoors *Social Conditions: Stress Time critical tasks	A	*ME: Device-user mouth proximity *AP: Rule -Based *AM: Enable speech modality	Services (B2B System)
[108] Turunen et al. 2009	*Speech *Motion G. *Extra Gestures *Indirect Manipulation	*Voice Commands *Tilt Vertical & down *Tilt Horizontal *Approach device to NFC tag *Key press	*Microphone *Accelerometer *NFC Reader *KeyPad	Smartphone (Symbian)	Yes *Vibra Feedback *Graphical Output	*Complementary Tilt Vertical & down + Key press (Move program selection) Tilt Horizontal + Key press (Zoom in and out) *Equivalence Voice Commands Approach device to NFC tag (Select menu options)	*Location Relative Location (Device-User mouth)	A	*ME: Device-user mouth proximity *AP: Rule-Based *AM: Enable speech modality	Entertainment (Media Center)
[120] Zaguia et al. 2010	*Speech *2D Gestures *Indirect Manipulation	*Voice Commands *Single Tap *Typing *Key press	*Microphone *Touch Display *Keyboard	Wireless M . Computer Mobile Standard PC (Laptop)	Yes *Audio Output *Graphical Output	*Equivalence (Select Fields) Key press Voice Commands Single Tap (Data Entry) Voice Commands Typing	*Location Home Work On the Go *Physical Conditions Noise Level: high, low Light Level: bright, dark, very dark	A	*ME: Noise level Location Light level *AP: Rule-Based *AM: switch user 's preferred modalities	Services (Flight Reservation)
[25] David et al. 2011	*Speech *2D Gestures	*Dictation *Typing	*Microphone *Touch Display	Smartphone (Android)	Yes *Audio Output *Graphical Output *Vibra Feedback	*Equivalence Typing Dictation (Write message, check answer)	*Social Conditions: Stress (cycling) *Location GPS absolute location	M	*ME: GPS speed *AP: Rule-Based *AM: Enable speech modality	Communication (AMC - SMS Application)
[56] Kong et al. 2011	*Speech *2D Gestures *Pen Gestures	(N/A)	*Microphone *Touch Display * Stylus	Smartphone	No *Audio Output *Vibra Feedback	*Equivalent <u>Outdoors</u> Speech 2D Gestures <u>Shopping Mall</u> 2D Gestures Pen Gestures *Assignment <u>Library</u> 2D Gestures	*Location: Outdoors Library Shopping Mall *Physical Conditions: Weather : sunny, cloudy, rainy Light level: bright medium, dark Noise level : low, high	F	*ME: Noise level, location light level, temperature weather *AP: Human-Centric (user's modalities preference score) *AM: Switch the set of current modalities	Entertainment (Social Networking Application)

Table 3.2: System-induced adaptation in mobile multimodal systems

Design and Development: Tools and Methods

Based on different adaptation mechanisms, specific frameworks and tools presented alternatives to design and build applications that automatically detect the modifications of the interaction context and adapt accordingly.

As part of the research project SMARTKOM [90], Böhler et al. [14] presented the first prototype of the mobile version of the Smartkom system. The relevance

of this paper relies on the introduction of a conceptual framework that deals with the flexible control of the interaction modalities as well as a new architecture for flexible interaction in the mobile setting. The framework supports that users as well as the system can initiate a modality transition between the default modality modes. The conceptual framework described five defined combinations of input and output modalities based on the user's level of attention within a car and a pedestrian setting, namely *default*, *listener*, *silent*, *car* and *speech-only*. The prototype version showcased transitions between modalities initiated by the user as well as initiated by the system. The adaptation mechanism is based on a set of predefined rules, for instance, a system-induced input transition in the driver environment allowed the pen gestures input modality to automatically switch off when the mobile device was connected to the car dock. Likewise, when high levels of background noise were sensed in the pedestrian environment, the speech input modality was switched off.

Using the same rule-based approach, David et al. [25] proposed a mobile middleware that facilitates the development and maintenance of mobile adaptive multimodal interfaces. The middleware is built upon the Android Framework and is composed of two layers including a *Services* and *Programming language* layer. The former is composed of two services that makes transparent for the developer the communication and acquisition of context information. The latter comprises a Java library that allows programmers to define situation contexts rules (conjunction and disjunctions on context variables) and, based on the validity of each of them, invoke the respective handler. One of the novelties of the approach is that this library is based on the context-oriented programming paradigm [94]. An instant messenger prototype was built to illustrate their approach. The application allowed users to read and write SMS messages using the keyboard or speech. The application adapted its input modes depending on the user's movements in a stressing condition like cycling. For example, when the user was riding a bike, the default input modality was automatically set to speech and when the user stopped the speech modality was deactivated.

In turn, Kong et al. [56] proposed a framework based on human-centric adaptation. In contrast with rule-based approaches, this paper quantifies the average user preference of a modality under an interaction context. For instance, a dark environment can reduce a user's preference score of modalities related to the visual display. Thus, adaptation can be seen as searching for an optimal set of modalities with the highest preference score for a given scenario. The adaptation algorithm also verifies that the selected modality does not exceed the system resource capacities. The adaptation algorithm is fired based on changes in the interaction context which encompasses user, device and environment properties. The application de-

sign and development process according to the framework can be summarised in three steps: Determining the tasks and available input/output for a device type. Then, the interaction scenarios should be determined as well as the interaction contexts. Last, but not least, the designers should evaluate the average preference score of a modality under an interaction context. To obtain this value, a survey with end users must be conducted. The results of the survey are used as inputs for the heuristic algorithm.

Applications

Zaguia et al. [120] presented an interesting approach for the detection of the optimal modality. Their research work explored the domain of web service access using a context-based modality activation approach. The important part of their approach is that it ensures that the invoked modalities are suited for a user's current situation. The system, more specifically a dedicated context interaction agent, is in charge of the detection of context information as well as the selection of the optimal modality. This value is calculated based on the evaluation of factors related to the interaction context. Interaction context refers to *user context* (e.g. regular user, deaf, mute, manually handicapped or visually impaired), *environmental context* (e.g. noise level or the brightness of the workplace) and *system context* (e.g. the computing device). The optimal modalities selected by the system have to meet two requirements. First, they have to be an appropriate modality based on the available mobile devices. Second, they have to be appropriate modalities based on the given interaction context. For instance, the speech modality is the optimal input modality if the microphone is found as available media. Then, in regard to the interaction context, the user should not be recognised as a hearing-impaired user. Likewise, the user's location should be recognised as on the go and the noise level should be mapped to quiet. This information as well as the parameters extracted from the user events are used as input for the multimodal fusion component. Based on these parameters, the multimodal fusion component decides whether the fusion is possible or not. This approach was illustrated using as sample application a ticket reservation system. The application allowed users to reserve a ticket using the optimal modalities provided by the system. Using PetriNet diagrams, all the transitions that could arise are depicted.

Porta et al. [85] investigated the use of multimodal input in the domain of decision support in order to ease the access to information in time critical situations. A business-to-business (B2B) system that supported 2D gestures, motion gestures as well as speech as input modality was developed. In this application, the speech and 2D gestures modalities were used in a complementary way to search infor-

mation, whereas motion gestures, in particular via shaking the device, were used for implementing an undo operation. In this approach, the system automatically switched on the speech modality when the user moved their arm holding the device closer to their mouth.

Within the entertainment domain, Turunen et al. [108] presented a multimodal media center interface. The interface allowed the users to interact using speech input, extra gestures (tangible interaction) and motion gestures. In the same fashion as Porta et al., the speech modality was automatically switched on when a user moved the device close to their mouth.

3.4 Analysis

This section focuses on analysing the data presented in Table 3.1 and Table 3.2. These summary tables encompass fourteen relevant research articles related to the study field of interest. The analysis is performed in terms of multimodal interaction, context influence and adaptation. The first two analyses serve as a conceptual basis to understand common multimodal composition patterns as well as to understand the suitability of modalities according to context variations. These aspects are paramount during the design phase of an effective multimodal context-aware adaptive application. The last analysis section focuses specifically on the core features related to system-induced adaptation.

This section is divided in three subsections, namely Combination of Modalities, Context Influence and Automatic Adaptation. It is also worth mentioning that the first two analyses are conducted with the entirety of the research work data. As previously described, all the selected research work present multimodal systems that take into account environmental influence. However, the analysis in the Automatic Adaptation section relies uniquely on the information corresponding of Table 3.2.

As observed in Table 3.2 and Table 3.3, the selected research work covers the whole spectrum of mobile devices described in Schiefer and Decker's taxonomy [96]. It is interesting to observe that before the appearance of modern smartphones like the iPhone in 2007, wireless mobile devices were mostly used to showcase and test research findings. Specifically, under this category, mobile standard PC and handhelds appeared in 8 out of 14 papers, which represents 57% of the articles. Only one article [62] made use of an Internet tablet. As of 2009, we observed a change in this pattern and researchers started to explore the new features of modern mobile devices and feature phones. Specifically, in 6 out of

14 articles we observed the usage of commercial smartphones running different types of mobile operating systems like iOS, Android or Symbian. Only in one article [92] we observed the usage of a feature phone. Interestingly, research work related to system-induced adaptation relied mostly on the use of smartphone devices.

The reviewed papers researched or showcased their work using proof of concept applications for different domains. Based on domain similarities, we have classified the research into four categories: *services*, *communication*, *map and GIS* and *entertainment*. The first category encompasses the search of information through different web services as well as form-filling tasks. Such applications were observed in 5 articles [112, 104, 89, 85, 120]. The remaining articles appear listed equally in the other three categories. Examples of applications observed in each category are a Shopping Assistant [112] (Services), Multimodal SMS Application [25] (Communication), a Multimodal Digital Book Player [89] (Entertainment) and a Tourist Guide [29] (Map & GIS). It is interesting to notice, that although health and education are key areas in mobile research, none of the articles proposed an application in these domains.

3.4.1 Combination of Modalities

This section provides an analysis of the modalities used by each project and outlines how different modalities were combined in terms of the CARE model.

From the fourteen articles, all the articles with one exception explored different equivalent and complementary combinations of modalities. The remaining article, Ronkainen et al. [92], explored the assignation of one specific input modality to perform a particular task. For instance, the authors studied the use of the *double tap* gesture at the back of the device as an alternative technique to silence the device or to start speech input recognition. This gesture was conceived to be used under contextual situations that restrict the use of the device's display. Under similar conditions, keyboard input with augmented vibration feedback when the user pressed a key was evaluated. From this article, it is important to notice the influence of the current input modality in the selection of the output modality. Analysing output modalities is out of the scope of this analysis, however this specific relationship has been reviewed. The results showed that in seven out of fourteen articles the selected input modality influenced the output modality [62, 26, 92, 14, 120, 25, 108]. It is also worth highlighting that only in Wassinger et al.'s work [112] the use of the redundancy property was observed.

Interestingly, results showed that speech input modality is present in around 86% of the articles with the exception of [86, 92]. This modality has been mostly explored in the form of voice commands and less explored using more complex speech recognition techniques like natural language dialogues [104, 85, 14] and dictation [29, 25]. Voice commands consist mainly of short phrases made up of few words and matched against a specified rule grammar. This interaction technique has been mostly used in conjunction with pointing-based techniques. In this way, it was possible to ask the system for information related to an element pointed by the user. Furthermore, some authors have used this interaction technique to map the default system's menu options to specific voice commands, as observed in [62, 26]. On the other hand, natural dialog in a broad sense can be seen as a human like conversation with the system. This type of interaction has been effectively used in car contexts as for example seen in Bühler et al.'s [14] work. Finally, dictation consists in the translation of spoken words into written text and was observed in the work presented by Doyle et al. [29] and David et al. [25].

Twelve out of the fourteen research projects made use of the touch displays of recent smartphones or PDAs. Modalities associated with this interaction sensor are 2D gestures and pen gestures, respectively with an appearance of approximately 64% and 50% in the reviewed papers. Although these modalities are very similar, they differentiated from each other since the former interacts with the display directly using the fingers and the latter relies on the use of a stylus. Hence, some specific interaction techniques are unique for each modality. Apart from this difference, the results showed that the interaction techniques used by the two modalities are very similar. In general, they were used for pointing tasks, specifically to select a specific item in the interface. Not commonly explored, yet interesting uses of these modalities are finger-based gestures and handwriting. Simple finger-based gestures, like strokes or symbols, were only explored in [62, 89]. The work presented in the last reference also allowed to draw symbols on the touch display using a pen/stylus. Handwriting, on the other hand, was only performed using a stylus in handheld devices in [112, 29]. Another modality that was observed with a high frequency among the articles, specifically with around 43% of appearance is indirect manipulation. This modality encompasses the interaction techniques performed with the device's physical keyboard or keypad. Mostly this modality appeared to be used in an equivalent manner with all the other modalities. Only Turunen et al. [108] used it in a complementary manner with motion gestures. Most of the existing research aimed to prove the added value of other modalities in comparison with the traditional keyboard and keypad interaction.

Less explored modalities are motion gestures and extra gestures appearing approximately in 29% and 14% of the articles. Motion gestures refer to gestures in thin air executed with the phone in hand and captured by accelerometers, compasses or magnetometers sensors. Rules or machine learning algorithms are used to recognise these different movements. Gestures retrieved in this way mostly use interaction techniques that allow to perform navigational or atomic tasks. For instance, navigation gestures like tilting the device up, down, left or right have been observed in three out of four articles [62, 86, 108]. In turn, atomic actions are mostly performed using a “shake” gesture. For instance, Port et al. [85] used this gesture to fire an undo action. The advantages of these type of gestures relied in their capability to enable one-handed interaction. Moreover, the level of attention to the device is also reduced.

Last but not least, the usage of extra gestures which refers to the interaction with tangible objects, has been only explored by Wassinger et al. [112] and Turunen et al. [108]. In the former, RFID tags were attached to products in a store and a user’s pick-up and put-back actions were evaluated to detect if the objects were either in or out of the shelf. On the other hand, Turunen et al. mapped the main options of the system to an A3 control board tablet that stored behind the menu options RFID tags. In both articles, user evaluations showed good acceptance rates from the participants. Table 3.3 shows a summary of the combinations between the aforementioned modalities.

	Speech	2D gestures	Motion gestures	Pen gestures	Indirect manipulation	Extra gestures
Speech		E: 62,26,56,120,25,89 C: 112,85	E:108	E: 26,29,14,89 C: 26,29,112,104,14	E: 26,120,89	E:108 C:112
2D gestures	E: 62,26,56,120,25,89 C:112,85		E:62	E: 112,26,56,89	E: 26,120,89	E:112
Motion gestures	E:108	E:62			E:86 C:108	
Pen gestures	E: 26,29,14,89 C: 26,29,112,104,14	E:112,26,56,89			E:26,89	E:112
Indirect manipulation	E: 26,120,89	E:26,120,89	E: 86 C:108	E:26,89		
Extra gestures	E:108 C:112	E:112		E:112		

Table 3.3: Modalities combination summary

As one can see, the combination of speech and on-screen gestures, either using a pen/stylus or finger-based interaction, has been well explored in the past few years. However, it is interesting to observe that the complementary combination of speech and pen gestures modalities is considerably higher compared to

using speech in conjunction with 2D gestures. In 5 articles [26, 29, 112, 104, 14] this type of combination was observed, whereas only two articles [112, 85] explored speech complemented with 2D gestures. This is probably due to the better accuracy provided by pen/stylus in comparison with a finger-based interaction. Coincidentally, the two references that made use of 2D gestures are not map-based applications where finer-grained precision is paramount. Besides the complementarity combination, equivalent combination of speech and pen gestures has been explored in four articles. Mostly this combination is used to provide the user with equivalent voice commands to access the application's menu options. However, one can notice that speech and 2D gestures have been used more frequently in this manner. Thereby, six articles used this type of modality combination with the same purpose. Finally, it is important to notice that pen gestures and touch gestures have been explored in an equivalent manner in four articles [112, 26, 56, 89]. The equivalent use of these two modalities is a current trend in commercial products such as the Google Note Smartphone, the Microsoft Office 2013 suite and the Windows 8 operating system.

It is also important to outline the common usages of the above mentioned combinations. In regard to the complementarity of speech and pen gestures, two articles [29][14] explored this combination for map-based applications. Bühler et al. [14] combined these modalities to improve the search task in a mobile map application. The system allowed to select a point in a map and then pronounce a sentence in natural language such as *I would like to know more about this*. Similarly, Doyle et al. [29] explored pen based gestures in conjunction with speech to perform two tasks within their tourist guide application. First, users could find the distance between two points in the map by uttering the command *Find Distance* and then drawing a stroke line on the map. Additionally, users were able to use dictation to create an annotation and assign it to a specific point of interest in the map. A very similar approach was presented by De Sa et al. [26]. The authors allowed to create annotations of a specific section in the Multimodal Digital Book using pen gestures and speech. However, in this application the user first selected a section of the book with the stylus and then uttered the voice command *annotate*. Another usage of this combination of modalities has been observed for the task of asking information for a particular item in a set of images or menulist options. For instance, within the shopping context described in Wassinger et al.'s work [112] the user was able to ask for information about a product using speech and pen gesture commands such as *What is the price of this*[pointing gesture]. The same usage was explored with 2D gestures in [112, 85] and with extra gestures in [112]. In these three references the use of fine-grained precision was not needed to achieve the task. As one can observe, complementarity is linked mostly with speech and pen or 2D gestures. However, it is worth to notice that Turunen et al.

[108] combined motion gestures with keyboard input. In this way, they allowed to expand the number of possible interactions permitted by the navigation keys. For instance, pressing the navigation keys while tilting the device horizontally allowed to zoom in different directions, whereas tilting the device vertically and downwards allowed to move between a user's programme selection.

Regarding equivalence combination, the added value of providing two or more equivalent modalities in an application relies on that the user is provided with multiple input options to access the main functionalities. For instance, combining speech with pen gestures, 2D gestures or traditional keyboard-based interaction is useful when the user's attention is not focussed on the device. Based on the same foundation, these modalities have been combined with motion gestures in [62, 86]. An interesting and useful usage of equivalent combination of modalities is pen with 2D gestures. For tasks like writing or drawing, the use of pen/stylus can have an added value over 2D gesture interaction. On the other hand, using 2D gesture modality permits the use of single-handed interaction for pointing tasks. Another interesting observed combination of equivalent modalities is the use of extra gestures (tangible interaction) as an alternative to speech ([108]). The authors highlighted that the interaction with real world objects can ease the use of complex devices for elderly people. For instance, in the context of the Multimodal Multimedia Center, users were able to select a program either by issuing voice commands like *I want to see all sports events* or by touching with the device the sports icons in the paper-based control board. The evaluated users preferred tangible interaction over speech, since the latter was more error prone. Similarly, extra gestures were used as alternative to 2D gestures and pen gestures in [112].

3.4.2 Context Influence

This section addresses the suitability of the above analysed input modalities according to specific context settings. This analysis relies on the findings from the articles presented in Table 3.1 and Table 3.2, particularly from the articles that conducted user studies in real world settings [62, 29, 89, 86]. Likewise, studies conducted in laboratory settings [112, 108, 56] are also taken into account because they evaluate the preferability or suitability of modalities in specific contexts. Finally, the context analysis performed in 4 articles [26, 92, 120, 14] are an additional resource to perform the present analysis.

In general, users feel comfortable using all modalities in private places, where social interaction as well as noise levels are low. For instance, in Wassinger et al.'s evaluations [112], the results showed that users feel comfortable using all modalities in private places, even speech. In this study, the studied modalities

were pen gestures, 2D gestures, extra gestures and speech. However, in public environments the results were different. The same result was observed in Zaguia et al.'s work [120], where all modalities were categorised as appropriate for the semantic location “home” with low levels of noise and a well illuminated environment. Last but not least, in Reis et al.'s work [89], a study performed in the wild, specifically in the four real settings *home*, *park*, *subway* and *car* showed that in the “home” environment, users are eager to experiment with modalities they found interesting. The results in this setting showed that all evaluated users used voice interaction for selecting menu options and also for data entry. This behaviour was not the same in the other three environments.

When ideal environmental conditions are altered, the user's preferred modality varies as well. Table 3.4 shows a summary of the suitability/preferability of each modality under specific environmental settings. The observed environment variables presented in the table are taken from the environment model described in the Cameleon reference framework [16].

Environment variables		Speech	2D / Pen /Indirect manipulation	Motion gestures	Extra gestures
Physical conditions	Brightness	B / D : 120	B : 120	B/D : 108	(N/A)
	Noise level	L : 29,120	L / H : 120	(N/A)	(N/A)
Social conditions	Stress	H : 62,14,89	L : 26,92,89	M : 86,62	(N/A)
	Social interaction	L : 89	M / H : 89,62,86	M : 86,62	(N/A)
Location	Semantic location	I : 89,29,112,120	I : 112,92,56,120,62	I : 62	I :112,108
		O : 56,62,14,89	O : 112,56,86	O : 86	O :112

Table 3.4: Modality suitability based on environmental conditions. The abbreviation **B** stands for bright and **D** for dark. The values that noise level, stress and social interaction variables can take are **L/M/H**, where **L** stands for low, **M** for medium and **H** for high. The semantic location **I** refers to Indoors and **O** refers to Outdoors

Speech

From the modality usage analysis, it was observed that speech is mostly used for specifying commands that map to system functionality or to ask for information combined with another pointing-based modality. However, independently from the task, the use of speech is preferable in environments where the physical conditions encompass low levels of environmental noise [29, 120] as well as low

levels of social interaction [89]. Speech is a modality that can also be performed in dark environments [120]. For instance, in the context study performed by Zaguia et al. [120], the use of speech was found appropriate under all types of light level conditions. However, in the same study, speech was only adequate to use in quiet environments. Similarly, based on their user study findings Doyle et al. [29] highlighted that the interaction time in noisy environments increased by 16.09% in comparison with quiet environments. The rise of the interaction time was attributed to an increase of user errors due to misrecognition problems. Thus, this modality has been found suitable for indoor environments such as *home* or *private places* [89, 29, 112, 120]. However, it was observed that when the user is “on the go” or in outdoors environments this modality is mostly suitable in the driving context [62, 14, 89]. Since the user is involved in a stressful situation like driving and the level of attention to the device is very low, interacting with speech is an optimal modality in this setting. It is interesting to notice that the algorithm presented by Kong et al. [56] to select the optimal modality chooses this modality for the context “on the go”. However, the authors did not specify under which particular outdoor conditions the modality was suitable. In regards to social influence, in Reis et al.’s study [89] it was highlighted that the use of speech in crowded environments is avoided due to privacy reasons and because the users felt embarrassed to talk in front of strangers. These results were observed in the tests performed in a park and subway.

Pen Gestures, 2D Gestures, Indirect Manipulation

Previously, pen gestures, 2D gestures and indirect manipulation modalities were analysed separately. However, in this section they are analysed together since all of them demand full visual attention to the device. Constraints related to noise level do not affect the use of these modalities, however very dark environments can constrain their usage [120]. Additionally, these modalities are found to be appropriate to use in indoor environments such as library/shopping mall [56], office [92, 120, 62] and home [120]. Moreover, in the user study performed by Lemmela et al. [62], users preferred to interact with 2D gestures instead of using speech in a walking context inside office areas. Likewise, the reported results from different user studies showed that users feel comfortable using them in public places as well [112, 56, 86]. However, when a task requires full attention of the user, it may require the use of another modality like speech [26]. Hence, these modalities are more appropriate in activities that generate a low level of stress. For instance, in Ronkainen et al. [92] the interaction with these modalities was established as difficult when the user is involved in tasks like bicycling, driving and in outdoor winter activities (due to the use of gloves). Furthermore, the interaction with these modalities in the train or in a busy street were categorised to have a medium level

of difficulty. In a similar environment, namely in a subway context, the results presented in Reis et al.'s study [89] showed that the users reported difficulties using only one hand and interacting with the device using 2D gestures. In these settings, users used only one hand since the other one was used to hold themselves safely in the subway. Finally, the influence of social interaction does not seem to limit the usage of these modalities. For instance, in environments that showed medium or high level of social interaction [89, 62, 86] users reported to feel comfortable using these modalities.

Motion Gestures

The usage of this modality has been observed among the reviewed articles as an alternative to the traditional key-based navigation using left, right, forward and back movements. According to field study results, the modality has been used with a good level of acceptance in outdoors environments [86] as well as in indoor environments such as office areas [62]. Since it does not require a high level of attention to the device, it can be suitable for medium level of stress situations where hand interaction is permitted. In both articles users were evaluated in a walking context and under a medium level of social interaction and stress. For instance, in Ramsay et al.'s user studies [86], the participants had to perform a task in conjunction with a partner and also answer calls during the evaluation. On the other hand, in Lemmela et al.'s study [62], users had to walk across different offices, stairs and doors. However, many of the participants from Ramsay et al.'s study mentioned that in a very stressful situation, such as a time constrained task, they would interact with the device's keypad instead. Finally, as one can notice, social interaction varied in both evaluations as well. However, this modality seems to be appropriate though. It is important to take into account that this modality can also be suitable for very dark environments as highlighted by Turunen et al. [108].

Extra Gestures

The usage of extra gestures (tangible interaction) was observed as an alternative input mode to access the application's functionality [112, 108]. In both articles, the usage of this modality was evaluated with users, however there is no available information regarding specific physical conditions (brightness, noise) and social interaction influence. Regarding the influence of location, in Wassinger et al.'s [112] study, users answered that they felt comfortable using this modality in public places as well as in private environments. Moreover, the study conducted by Turunen et al. [108] was performed in a controlled environment simulating a home-like setting. The results under this setting showed that users found appro-

priate the usage of this modality for that context environment. Additionally, the authors highlighted that extra gestures is an intuitive, simple and secure interaction technique which requires only little cognitive load. Hence, it could be useful in stressful situations where the attention is not centered on the device.

3.4.3 System-Induced Adaptation

Based on the research work presented in Table 3.2, an analysis of the system-induced adaptation core components is presented. First, we provide an analysis of the *monitoring entity* component. Therefore, we reviewed the entities that monitor and start the adaptation mechanism in the reviewed articles. Relying on the Cameleon’s framework environment properties classification (physical conditions, location and social conditions), it was possible to analyse which specific environment variable was used for which task. Furthermore, we analysed the type of *adaptation policy* and *adaptive mechanism* technique observed in the articles. Table 3.5 provides a summary of the articles corresponding to each core component.

Monitoring Entity			Adaptation Policy		Adaptive Mechanism	
Physical Conditions	Location	Social Conditions	Rule Based	Heuristic Algorithms	Enabling	Switching
Acceleration: 14,25	Relative position: 85, 108	(N/A)	14 25 85 108 120	56	108 85 25	14 120 56
Noise level: 14,120,56	Absolute location: 25					
Light level: 120,56	Semantic location:					
Temperature: 56	14,120, 56					
Weather: 56						
Time: 56						

Table 3.5: System-induced adaptation core features

Monitoring Entity

This section answers one of the questions formulated at the beginning of our study, namely to which conditions these types of systems adapt and specifically which environment variables call for which adaptation. One can observe from Table 3.5 that only physical and location-based variables were used as adaptation triggers. Interestingly, none of the reviewed papers use methods for social cues detection based on built-in mobile sensors or other techniques able to detect social interaction. Detailed information about the *physical conditions* and *location* variables that influenced the adaptation are described below.

Physical Conditions: The monitoring entities under this category are mostly built-in sensors that constantly sense changes in physical conditions like noise and

light level, weather or acceleration. Large variations in these values lead to a possible input modality adaptation. For instance, in three articles the usage of noise level sensing [120, 56, 14] was observed. However, this information is mostly used in conjunction with information derived from other sensors. The first two references [120, 56] used noise data in conjunction with the results obtained from the light sensor and location information to trigger an adaptation call. Only in [14] it was used as the only discriminant to fire an adaptation call. In this article, particularly in the pedestrian environment, high levels of noise were monitored to turn off the speech input modality. It is also important to notice that in the first two articles, the raw values from noise and light level are mapped to semantically higher information such as *quiet* and *noisy* for noise level data and *bright*, *dark* and *very dark* for light level data.

Apart from sensing noise level variations, Bühler et al. [14] and Lincoln et al. [25] sensed acceleration variations to identify when an object is still or moving. In the former, it was used in a driving context where the speed of the car was constantly measured using a car-PC running a specialised software connected with a CAN bus¹ to access the state of the car. In the latter, it was used in a cycling context, where the speed value from the GPS sensor was captured. If this information was not available, the location was stored in a history queue to then calculate the average speed using distance and time values. Finally, only in the framework proposed by Kong et al. [56], variables such as temperature, weather and time were considered as possible adaptation triggers. The authors mentioned that any change in the values of these variables fires the adaptation algorithm.

Location: The observed monitoring entities under this category are clustered in three groups, namely *relative*, *absolute* and *semantic location*.

Relative location refers to the location of the device or user in relation with another point of reference. For instance, Turunen et al. [108] and Porta et al. [85] monitor the proximity of the device to the user's mouth to trigger an adaptation call. Specifically, it was monitored by recognising a specific arm gesture where the user brings their arm and device closer to the mouth. This type of gesture recognition is made using the built-in accelerometers from the device.

On the other hand, *absolute location* refers to the exact geographic position measured with latitude and longitude coordinates obtained from the GPS sensor. As previously mentioned, only Lincoln et al. [25] used this information when the speed value from the GPS (Global Positioning System) was not available. It is worth to notice that three articles [14, 120, 56] use semantic locations as

¹http://en.wikipedia.org/wiki/CAN_bus"

adaptation triggers. Semantic locations refer to places that convey a meaning to everybody but not specify a particular geographical location, such as home. For instance, Bühler et al. [14] monitored whether the user was in the car by sensing when the handheld device was connected or disconnected to the car's docking station. The change of location triggered an adaptation call. Zaguia et al. in [120] analysed three semantic locations: *home*, *work* and *onthego*. Similarly, Kong et al. [56] use as semantic locations for their user evaluation the *outdoors*, *shopping mall* and *library* environments. However, it was out of the scope of the research work to describe how the system detected the user's current semantic location.

Based on this information, it can be generalised, that since semantic location does not vary very often, dynamic variables like noise level and light level are associated to each location to call for an adaptation. Hence, an adaptation is fired when the location is changed and also when inside one semantic location the environmental conditions are altered. For instance, if the monitoring entity recognised that the user is in an *indoor* location, the adaptation policy component, based on the noise level information can activate one input modality. However, if the sensors recognised that the user is *outdoors*, under the same noise level conditions, another modality can be activated. The decision relies on the adaptation policy component.

Adaptation Policy and Adaptive Mechanism

This section describes how the input adaptation takes place in terms of the *Adaptation Policy* and the *Adaptation Mechanism* components. In regards to the *Adaptation Policy* component, we analysed which decision inference mechanism (rule-based or heuristic algorithm) was mostly used among the reviewed articles. Regarding the "Adaptive Mechanism" we analysed the possible modifications that end users could perceive after an input modality adaptation was triggered. Hence, we reviewed whether the modifications occurred by enabling and disabling a modality or by switching between a set of modalities.

The rule-based approach refers to simple logical rules that indicate when the adaptation has to take place as well as what kind of adaptation should take place. One can easily notice that all reviewed articles with the exception of Kong et al.'s work [56] followed this approach. In most of the reviewed articles, very straightforward logic rules were defined, mostly to activate and deactivate one single modality. It is worth to notice that three out of six papers [108, 85, 25] focussed on enabling and disabling the speech modality automatically without forcing the user to press the *speech to talk* button. As previously mentioned in [108, 85] the proximity of the device to the user's mouth was monitored. One single rule was

defined in these two articles. The rule verified if the gesture performed by the user was recognised as the user intending to bring the phone close to their mouth. If the gesture was recognised, the speech modality was automatically activated. Although in both articles other modalities were available, this rule only affected the speech modality behaviour. Using the same adaptive mechanism technique, Lincoln et al. [25] defined a rule that verified if the speed value captured from the GPS sensor went over a threshold ($\text{location.speed} > 5$). If so, the speech modality was activated, otherwise deactivated.

On the other hand, three articles [14, 120, 56] used as adaptation mechanism the *switching* technique. For instance, in Bühler et al. [14] this technique was used to switch between the five input interaction modes. Each interaction mode encompassed a set of allowed and suitable modalities for specific contextual situations. When a rule was satisfied, the suitable interaction mode was activated. For instance, one rule specified that if the user's current location was different from car, the default interaction mode that relied on speech and pen gestures was activated. In turn, when the noise level was sensed too high, the system switched to the listener mode and deactivated the speech input modality.

Similarly, Zaguaia et al.'s work [120] relied on a more complex set of rules to obtain the set of suitable modalities for a particular context and device. Hence, when context parameters changed, the system evaluated again which set of modalities fit the new environment parameters. Rules based on the conjunction and disjunction of environment properties are defined for speech, 2D and pen gestures as well as for indirect manipulation modalities. For instance, the speech modality is said to be optimal when the available media is *microphone* and *speech recognition*, as well as when the user's location variable is not set as *on the go* and the noise level variable is not labelled as *noisy*. Similarly, the heuristic algorithm presented by Kong et al. [56] selects the set of modalities that achieves a maximum preference score.

As mentioned above, Kong et al. [56] was the only article that reviewed another approach in regard to the *adaptation policy* component. The authors argued that rule-based approaches face some issues, for example they do not cover all interaction scenarios and rules might conflict with each other. Hence, they proposed a human-centric adaptation approach using a heuristic algorithm. The term heuristic is used for algorithms which find solutions among all possible ones. Hence, in this context, the objective of the adaptation algorithm was to find a set of modalities that achieves a maximum preference score. The preference score for each modality is designated based on the average value obtained based on a user survey done with 259 participants.

The results from the algorithm designated for the “outdoors” location the following modalities: 2D gestures and speech whereas for a *shopping mall* location 2D gestures and pen gestures and for a *library* location only 2D gestures were selected.

3.5 Guidelines for Effective Automatic Input Adaptation

During the analysis phase, we noticed that the field of system-induced adaptation has barely been addressed. The analysis of user-induced adaptation showed that existing research efforts provide the necessary conceptual basis to systematically design a context-aware adaptive application. However, these concepts have only been taken into consideration by two references [56, 120].

Therefore, based on the reviewed articles and the results obtained from the analysis sections, we proposed a set of guidelines that can be used to design a context-aware adaptive and multimodal mobile application. These guidelines unify the key aspects and stages observed in the reviewed articles. Additionally, for each guideline, we analysed how it supported the guidelines we reviewed in the background studies section.

We organised the guidelines into three main phases: *Context and modalities suitability analysis*, *Multimodal tasks definition* and *Adaptation Design*. It is worth to notice that these guidelines are targeted to system-induced adaptation of input channels.

Context and Modality Suitability Analysis

Prior to the multimodal and adaptive design, the influence of environment factors should be evaluated. In this phase two activities are important: conduct a context analysis and define suitable modalities for each semantic location.

1. **Conduct a context analysis:** It is important to define in advance the semantic locations (e.g. park, car, street or office) where the user is mostly going to interact with the application. After defining these locations, a context analysis should be conducted to have an overall picture of the environment factors that influence each location. Specifically, for each location, the influence of environment variables such as noise level, social interaction or stress should be analysed. Then, qualitative values like *low*, *medium*, *high*

should be assigned to each location/environment variable pair. This guideline supports the analysis of mobile scenarios reviewed in the conceptual frameworks [62, 26, 92, 56].

2. **Define suitable modalities for each semantic location:** Based on the physical and social conditions assigned to the semantic locations, the designer should specify which modalities are appropriate for each location. In this way, decisions about which modalities should the application support, can be taken. To achieve this, the interaction easiness for the pair location/modality should be evaluated. For example, if the context analysis of the semantic location *street* outputs the following values: noise level (*high*), social interaction (*medium*) and stress level (*medium*), then the speech modality is labelled with a *difficult* value and the modality is considered as not suitable. The designer can rely on the findings from the section 3.4, to assign these values. This guideline supports the guidelines reviewed in section 2.2 (MU-G2) and section 2.3 (M-G2.7 and M-G2.6).

Multimodal Tasks Definition

After defining the modalities that the application will support, this phase encompasses the design of the multimodal input channels. It includes two guidelines, namely the selection of multimodal tasks and the definition of equivalent input modalities.

1. **Select tasks that will support a multimodal behaviour:** It is important to specify which tasks in the application will support a multimodal behaviour. Thereby, ideal multimodal tasks are frequently used tasks, error-prone tasks or tasks that involve a complex process. This guideline supports the guidelines reviewed in Section 2.2 (MU-G1) and Section 2.3 (M-G1.1 and M-G2.5).
2. **Define equivalent modalities for the multimodal tasks:** For any given multimodal task, an interaction technique must be specified for each available modality. In this way, the user can perform the same task using any of the supported modalities. This guideline supports the guidelines reviewed in Section 2.2 (MU-G6) and Section 2.3 (M-G2.9).

Adaptation Design

Based on the the *context* and *modality* analysis information obtained from the previous stages, the designer should specify the design of the adaptation process. Three aspects should be taken into account in this phase, namely the definition of

adaptation triggers and monitoring entities, the definition of the adaptation policy and modalities and context status feedback.

1. **Define adaptation triggers and monitoring entities:** In this step, it must be clear for the designer which type of environment variables will influence the adaptation (physical conditions, social conditions, location). Likewise, it should be specified how the environment data related to the variables is going to be captured. Moreover, it should be specified how this sensor-data is mapped to meaningful information for the application. This guideline supports the guideline reviewed in Section 2.3.3 (CA-G1) and Section 2.4 (SA-G1).
2. **Define adaptation policy mechanism based on context analysis:** Independently of the selected adaptation policy mechanism (rule based or using an heuristic algorithm), the assignation of input modalities according to the changes in the environment values should be defined. These design decisions should take into account the *Context analysis* and *Multimodal Tasks Definition* performed in the previous stages. This guideline supports the guideline reviewed in Section 2.4 (SA-G3).
3. **Define adaptation mechanisms:** In this type of adaptation, the designer should decide between two possible adaptation mechanism: enable/disable one specific modality or automatically switch a set of modalities. This guideline supports the guideline reviewed in Section 2.4 (SA-G2).
4. **Provide modality and context status feedback:** The designer should provide means to display the available modalities and the current context status. The status should be visible all the time for the users, but not in an obtrusive way. This guideline supports the guidelines reviewed in Section 2.2 (MU-5), Section 2.3 (M-G2.2) and Section 2.3.3 (CA-4 and CA-3).

Chapter 4

Analysis, Design and Implementation of an Adaptive Multimodal Agenda

In the previous chapter, we reviewed and analysed the field of input channels adaptation in multimodal mobile systems. The findings from the reviewed articles as well as the results from the analysis sections converged towards a set of design guidelines.

Based on these findings, this chapter describes the underlying process behind the design and development of an adaptive multimodal mobile agenda application.

4.1 Motivation

The set of guidelines proposed in the previous chapter provides an insight and a systematic approach to design mobile multimodal adaptive interfaces. However, another paramount aspect of multimodal systems is their fusion algorithm. Interestingly, only one reference from the reviewed articles [120] addressed how the fusion engine and adaptation mechanism should communicate with each other. Nonetheless, it was not described at what level the fusion was executed or which type of architecture was used.

Hence, the present proof of concept application aims to explore and describe from a design and implementation perspective the whole process. Furthermore, the less explored modalities highlighted in Section 3.4.1 are selected as the supported input resources for the application. In this way, we intend to investigate the use of new combinations of modalities.

4.2 Analysis and Design

The Multimodal Adaptive Agenda (MAA) is the proof of concept application proposed in this thesis. MAA is a mobile calendar application enhanced with multimodality and adaptive capabilities. The application is situated in the domain of emphServices and particularly explores multimodal form-filling tasks.

Frequently used calendar tasks like, emphcreating a new event, entail a cumbersome process. For instance, as observed in Figure 4.1, to schedule a new calendar event the user has to follow a three step process just in order to reach the form filling window.

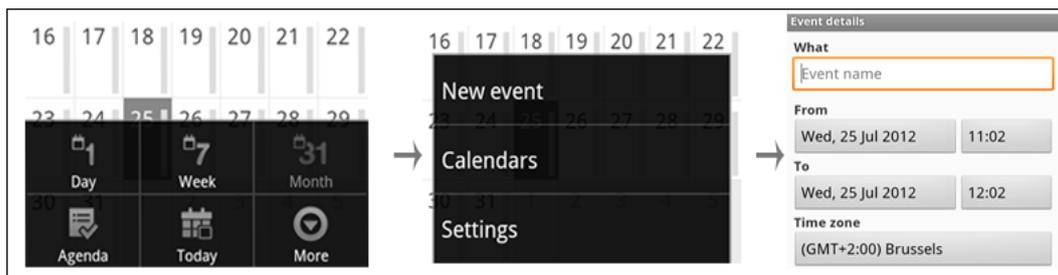


Figure 4.1: Three step process for creating a calendar event

Therefore, in modern Android and iOS based devices, the use of speech is explored to ease these tasks. Calendar events can be set up using voice-based assistants like GoogleNow¹ and Siri². However, as reviewed in the analysis of Section 3.4.2, speech is not appropriate to use in all the mobile settings. Hence, in MAA, the use of alternative modalities in conjunction with system induced input adaptation is explored.

Following the three phases described in the proposed design guidelines, namely *context and modality suitability analysis*, *multimodal tasks definition* and *adaptation design*, the supported input modalities, interaction techniques and adaptation rules are described.

¹<http://www.google.com/landing/now/>

²<http://www.apple.com/ios/ios6/siri/>

4.2.1 Context and Modality Suitability Analysis

The first step in the design of MAA was the context analysis of the locations where users will most likely use the application. Two indoor locations were evaluated as well as an outdoor one, namely *Home*, *Work* and *Street*. Then, the influence of three environment variables in each location was evaluated. The environment variables *noise level*, *social interaction* and *stress level* were selected to evaluate each location. During the analysis in Section 3.4.2, it was noticed that the variations in these three parameters were determinant to influence the usage of a modality in a specific setting. Then, the level of influence was assigned using a high/medium/low interval scale. In this way, Table 4.1 illustrates the results of this analysis. These values provide us with an overview of the context influence.

	Noise level	Social interaction	Stress level
Home (sitting)	Low	Low	Low
Work (sitting at desk)	Medium	Medium/High	Medium/High
Street (walking)	Medium/High	Medium	Medium

Table 4.1: Context analysis

This information serves as a basis to evaluate how easy or difficult it will be to use a specific modality at a particular location. The modalities that should be analysed are the ones supported by the device on which the application will run. Since we decided that the application should run in modern smartphones, the available input modalities are speech through the built-in voice recognition engine, 2D gestures using multi-touch displays, motion gestures using the built-in sensors such as accelerometer and extra gestures using the built-in near field capability.

The suitability level of each input modality was evaluated for each semantic location. For this analysis, a different three level qualitative scale (*easy*, *medium*, *difficult*) was used. In this way, if the interaction appeared to be very difficult for a particular modality/location pair, the modality was considered as not suitable. In turn, when the modalities were evaluated with an *easy* value, the modality was considered as suitable.

Table 4.2 provides a summary of the values obtained after analysing all input modalities. To assign these values, the findings from Section 3.4.2 were taken into account. For instance, the home/speech pair was marked as easy, since at that location the values of noise level were categorised as commonly low, as well

as the level of social interaction and stress. However, in the *Street* location the interaction was categorised as *difficult* since the values of social interaction and noise level were previously categorised as medium and high. On the other hand, the modality 2D gestures was set to *medium* within the *Work* and *Street* location since the stress level was set to *medium/high* in these locations. Thus, the level of attention to the devices is reduced and the interaction becomes more difficult. Finally, motion gestures and extra gestures were labelled with an *easy* value since neither of them are negatively affected by high levels of noise, social interaction or stress level.

	Speech	2D gestures	Motion gestures	Extra gestures
Home	Easy	Easy	Easy	Easy
Work	Difficult	Medium	Easy	Easy
Street	Difficult	Medium	Easy	Easy

Table 4.2: Ease of use of different input modalities according to context

This information was used to take decisions regarding the modalities and interaction techniques to be used. For instance, we noticed that the use of speech was only appropriate within the home environment. Hence we decided not to include it as a supported modality in the proof of concept application. Then, it was noticed that the 2D gestures modality has a *medium* value of suitability in two settings. Therefore, apart of the single tap interaction, we decided to explore 2D gestures interaction techniques that demand less attention to the device.

Hence, based on this analysis and the motivation to use the less explored modalities found in the Section 3.4.1, the application supports 2D gestures (single tap and symbol-drawing), motion gestures and extra gesture modalities.

4.2.2 Multimodal Task Definition

Functional requirements specify all the functionality supported by the system. However, not necessarily all the requirements should support a multimodal behaviour.

Therefore, and taking into account the guideline ‘*Select tasks that will support a multimodal behaviour*’, in MAA four tasks support a multimodal behaviour.

- ▷ Create new events
- ▷ Save and cancel the creation of new events

- ▷ Move back and forth through the months in the calendar
- ▷ Move the start date of an event back and forward by one day.

Additionally, based on the guideline ‘*Define equivalent modalities for the multimodal tasks*’, the specific interaction techniques related to each modality and task were defined. From Table 3.3 we noticed two equivalent combinations that were not explored: motion gestures with extra gestures and motion gestures with 2D gestures (symbols drawn in the touch display). Thus we explored these combinations. Additionally, all the tasks can be performed using single tap interaction.

	2D gestures		Motion gestures	Extra gestures
	Single tap	Symbol drawing		
Create new event	"New" button		"Shake" gesture	Approach NFC tag
Save event	"Save" button	"CheckMark" symbol	"Shake " gesture	
Cancel event creation	"Cancel" button	"Line" symbol	"Face Down" gesture	
Move between calendar months	"Left/Right" button		"Flick Left / Right" gesture	
Change day (Date Dialog)	"+" or "-" button		"Flick Up/Down" gesture	

Table 4.3: Supported input modalities and interaction techniques

4.2.3 Adaptation Design

As previously described in Table 4.3, the user is able to interact with the application using four interaction techniques. However, users might feel overloaded by having all the input modalities available at once and having to decide which one to use. To address this issue, the application defines a default modality (2D gestures) and incorporates supporting modalities according to the context conditions.

As recommended in the guideline ‘*Define adaptation triggers and monitoring entities*’, the possible adaptation triggers were identified. Based on the information from Table 3.5, *semantic location* and *noise level* were selected as the variables that call for an adaptation. To facilitate the recognition of semantic locations, only the change between *indoor* and *outdoor* is tracked by the monitoring entities. Social interaction is not a variable of our interest, since all supported modalities showed a high level of social acceptability according to the findings from the analysis presented in Section 3.4.2.

Although noise level variations do not affect the recognition of any of the supported modalities, variations in these values are correlated with the increment in the user’s stress level [116]. According to Evans and Johnson [37], not only loud or sudden noises provoke a stress response. But even low levels of noise can have a potentially stressful effect. Therefore, these variations are then important to consider, since in Table 4.2 we remarked that medium or high levels of stress make the interaction with 2D gestures modality difficult (related to work and street locations).

Then, if the user is located in an indoor environment and the noise level is categorised as medium or high by the application, the user might be involved in a stressful situation. Hence their attention level to the device might be affected. Therefore, when the monitoring entities sense this level of noise, an adaptation call is triggered and modalities that demand less attention to the devices are available to use.

Finally, taking into consideration the guideline ‘*Define adaptation policy based on context analysis*’, the different transitions of modalities were specified. This information is the conceptual basis to define context rules.

Table 4.4 outlines the set of modalities corresponding to the indoor location and different values of noise level. As observed, when the level of noise increases, new modalities are added. It is important to notice that the user’s activity is monitored as well. Although this variable is not an environment variable, it is monitored to avoid false positives when using motion gestures.

Thus, when the system detects that the user is walking, motion gestures do not appear in the set of suitable modalities.

Location	Noise Level	User Activity	
		Still	Walking
Indoors	Low	Extra gestures	Extra gestures
	Medium	Extra gestures Motion gestures	Extra gestures (N/A) 2D gestures (symbol drawing)
	High	Extra gestures Motion gestures 2D gestures (symbol drawing)	Extra gestures (N/A) 2D gestures (symbol drawing)

Table 4.4: Indoor locations: supported input modalities

The same analysis was performed for the outdoor location. However, taken into consideration that in outdoor environments the user might be exposed to more distractions, we decided to provide more supportive modalities even if the noise level is set to *low* and *medium*. This assignation can be observed in Table 4.5.

Location	Noise Level	User Activity	
		Still	Walking
Indoors	Low	Extra gestures	Extra gestures
	Medium	Extra gestures Motion gestures	Extra gestures (N/A) 2D gestures (symbol drawing)
	High	Extra gestures Motion gestures 2D gestures (symbol drawing)	Extra gestures (N/A) 2D gestures (symbol drawing)

Table 4.5: Outdoors locations: supported input modalities

4.3 Architecture

Independently from any specific technology, the top level architecture of MAA is depicted in Figure 4.2. As one can observe, the multimodal and adaptive components receive two external sources of information. The first comes from the events triggered by the user when using the supported input modalities. These events are recognised and processed by the *Modality Recognisers* component. On the other hand, the second source of information comes from the environment for example in terms of noise or a user's location. This information is constantly tracked by the *Entity Monitoring* component.

Then, this semi-processed information is sent to the *Multimodal and Adaptive Handler*. The *Fusion Engine*, *Adaptive Mechanism*, *Adaptation Policy* and *Dialog Manager* constitute this component and are responsible of the following tasks:

- ▷ *Fusion Engine*: Receives the information processed by the input modality recognisers. This component communicates with the *Adaptive mechanism* component to obtain the modalities that are allowed for the current context. Relying on this information, the fusion engine provides an interpretation of the user's intent and a notification is sent to the *Dialogue Manager* component.
- ▷ *Adaptive Mechanism*: This component is in charge of handling the adaptation calls that are sent by the monitoring entities. Then, based on the

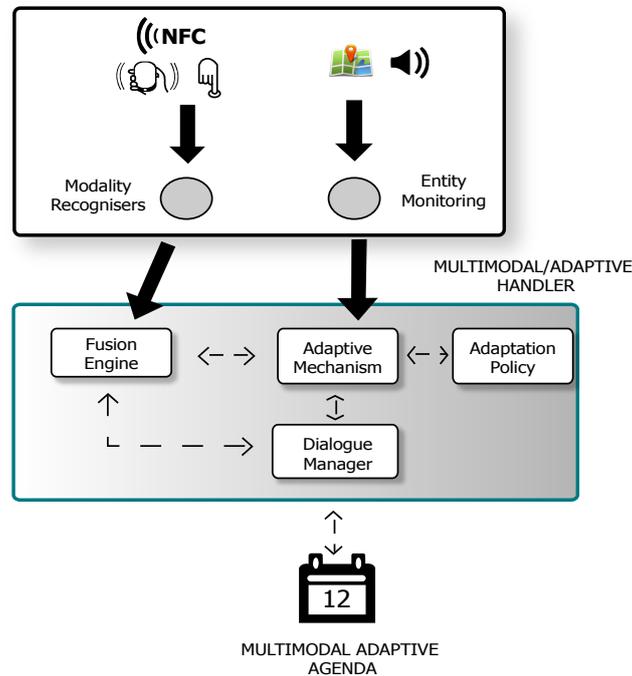


Figure 4.2: Top level architecture

information provided by the *Adaptation Policy* component, it notifies the *Dialogue Manager* when a change occurs.

- ▷ *Adaptive Policy*: Manages the context rules defined in the application and evaluates the current context information against these rules. This component notifies the *Adaptive Mechanism* component which rule satisfied the context parameters.
- ▷ *Dialogue Manager*: This component receives notifications from the *Fusion Engine* and *Adaptation Mechanism*. Based on the specific messages send by these components, it updates the user interface of the Multimodal Adaptive Agenda.

4.4 Technology

4.4.1 Android

Android is an operating system for mobile phones and tablets devices. The first release of the system appeared in 2008 and was developed by Google in union of the Open handset alliance, a consortium of firms with the purpose to develop open standards for mobile devices. Android was built with the intention to be the first open source platform for mobile devices. Eight versions of the operating system have been released until the latest Jelly Bean³ version.

General Architecture

Android's architecture is structured in five layers; each of them provides a higher level of abstraction to interact with the hardware and operating system. As shown in Figure 4.3, the first layer comprises the Linux Kernel, hence Android is built on top of Linux and some advantages hinge on this design decision:

- ▷ Android can run on different type of hardware avoiding the vendor lock in problem.
- ▷ Linux is well known as a secure operating system; hence Android relies on the Linux security model, especially in the permission model.

The second *Libraries* layer, includes a set of system libraries that will be used for the application framework layer and the Android runtime. The Android runtime includes the Java core libraries and the Dalvik Virtual Machine. This virtual machine was designed specifically for Android by Google and is also open and licensed free whereas the Java Virtual Machine is not.

The *Application framework* provides specific Java libraries that allow the developers to create Android based applications. Finally, the *Applications* layer contains the built-in applications from the operating system such as Contacts, Browser or Calendar.

Android SDK

Since Android is an open source project, developers have access to all the platform source code. If needed, they can change part of the base code and there is no need to contract any type of license. Also an SDK is provided to offer developers

³<https://developer.android.com/about/versions/jelly-bean.html>

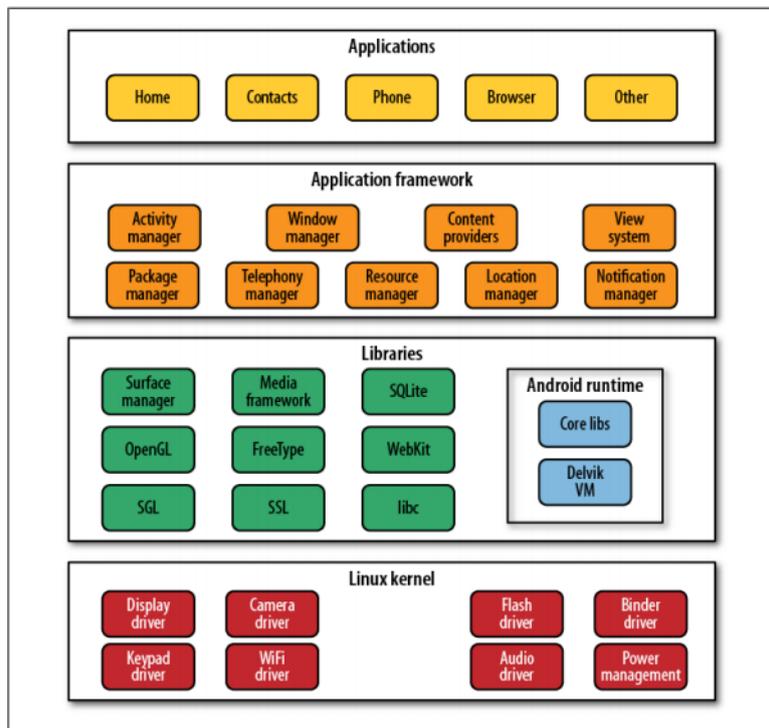


Figure 4.3: Android stack. Image taken from [40]

the necessary tools to start building an Android application.

The Android programming language is Java, therefore most programmers that are familiarised with this language will have a low learning curve. However Android does not use the Java Virtual Machine to run the generated byte code. In this case a different virtual machine called, *Dalvik Virtual Machine*. After compiling the code, the source code is packaged in an archive with the extension apk. This apk file is the application itself that Android-based devices install and run.

Eclipse is the official and most popular Integrated Development Environment (IDE) that supports Android application development. However, an additional plug-in is needed to this end. The Android Development Tools (ADT) have to be installed in order to extend the IDE functionality. This plug-in allows to manage Android projects and also includes the Android Virtual Device Manager that permits to create different device emulators for the different available versions of the API.

Android Framework Components

The Android framework defines four important components. These components are the key pieces to build any Android application. A brief description of each of them is listed below:

Activities: The Activity class is in charge of creating a window where different UI components can be placed. An application is composed of several activities.

Services: A Service is an application component that allows the developer to perform expensive operations in the background.

Content Providers These classes allow the developer to retrieve and share data with other applications. Android provides some built-in content providers such as Contacts or Calendar. Any application can query this information and also modify it.

Broadcast Receivers These components enable the capturing of messages from the system, for example if the battery is low or when the system boots. The receiver is a class that handles what happens when a particular event occurs.

4.4.2 Near Field Communication

Near Field Communication (NFC) is a set of short-range wireless technologies. It requires a distance of 4 cm or less to initiate a connection. Unlike Bluetooth, NFC does not require pairing, hence the connection is faster [1]. This technology has been used in different types of domains, for example payments (Google Wallet), public transportation (mobile ticketing), entertainment (Smart posters and event information) and business (business cards exchange).

NFC permits to share small payloads of data (up to 32kB) between an NFC tag and an NFC enable device, or between two NFC-enabled devices. Figure 4.4 shows how NFC tags can be embedded in a variety of products like printed or custom stickers, cards, wristbands or key-fobs.

NFC tags consist of data encoded in the NFC Data Exchange Format (NDEF). NDEF is a light weight binary standard for data exchange defined by the NFC forum.⁴ NDEF data is encapsulated inside a message (NdefMessage) that contains

⁴<http://www.nfc-forum.org/home>



Figure 4.4: NFC products

one or more records (NdefRecord). As observed in Figure 4.5, each record is described by a header and the payload. The header specifies a type, a length, and an optional identifier. The type of the identifiers may be URIs, plain text, MIME media types, or NFC-specific types [7].

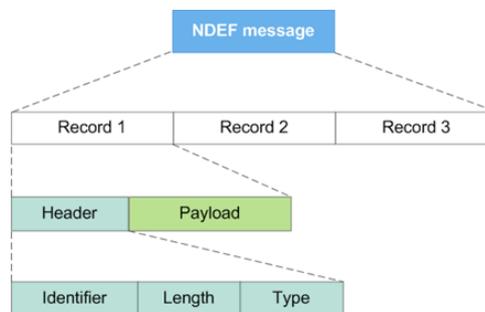


Figure 4.5: Ndef record. Image taken from [3]

In Android 2.3, Google introduced the Reader Mode NFC functionality. Starting with Android 2.3.3 (API level 10), the ability to write data to an NFC tag and exchange data via P2P mode is also available [70].

4.5 Implementation

Figure 4.6 highlights the implementation of the top level architecture using Android SDK framework components. As one can observe, the communication between the Android Views, Model and Activities rely on the Model View Controller (MVC) design pattern. Hence, the Views (*Android Layouts*) deliver user input events to the Controller (*Activities*). The Controller makes modifications to the Model (*EventDAO class*) and responds to the user events modifying the corresponding view components. The interaction between these components allow to implement the agenda functionality described in Section 4.2.2.

However, the capability to support multiple input modalities fusion and adaptive behaviour is achieved through the components from the *Multimodal and Adaptive Handler*.

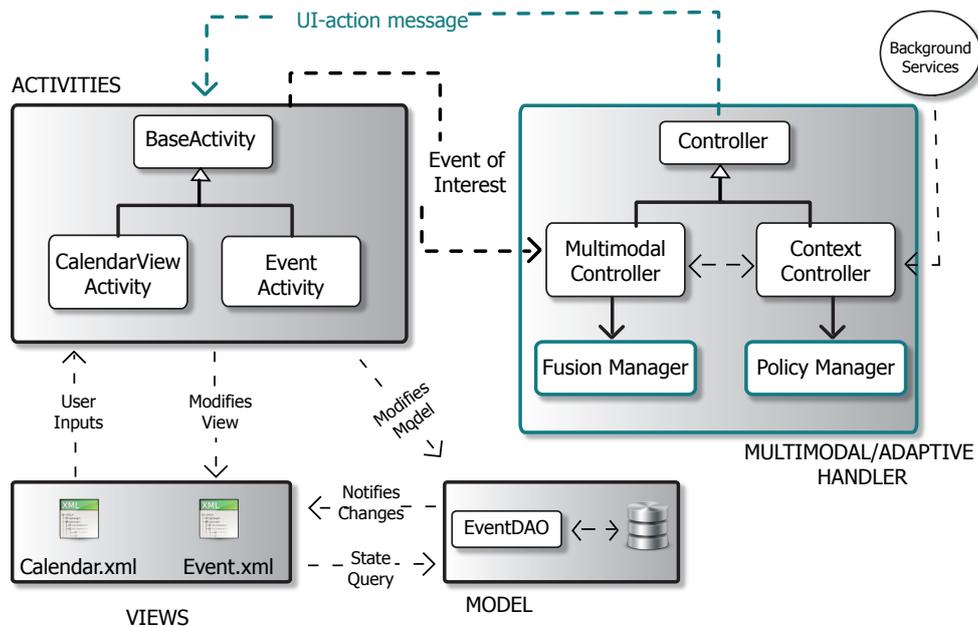


Figure 4.6: Android-based implementation of the top level architecture

Thereby, user events are not processed directly by the Activities, rather they are sent to the `MultimodalController`. When an event of interest is received, this component requests the `ContextController` the modalities permitted for the current context. Then, using this information, the `FusionManager` component interprets the user input and sends an UI-action message to the Dialogue Manager (`BaseActivity`). At this moment, the interface updates accordingly.

It is important to notice that the `ContextController` does not depend on any `MultimodalController` action. When an adaptation call is fired by the `BackgroundServices`, the `ContextController` handles this call by requesting the `PolicyManager` component the set of modalities associated with a matched context rule. Based on this decision, the adaptation mechanism notifies the DialogueManager about the change. Then, the interface updates the current context information as well as the set of permitted modalities.

The following sections describe implementation details about the components described above. First the *Views* and associated *Activities* components are described to illustrate MAA’s user interface. Moreover, a section is devoted to explain how the input modalities are recognised by the *Activities* and further processed. Likewise, the Fusion Engine component (MultimodalController and Fusion Manager) as well as the Adaptive Mechanism (ContextController) and Adaptation policy (PolicyManager) are explained as well.

4.5.1 Views and Activities

As previously explained, the user interface of the application is handled by the *Views* components, referred in the Android SDK as layout components. Each layout is related to a concrete Activity. The Multimodal Adaptive Agenda (MAA) has two main layouts, namely the *calendar.xml* and *event.xml*. These layouts are associated with the *CalendarViewActivity* and *EventActivity*.

Figure 4.7 depicts the result of rendering the aforementioned views. Important sections from the user interface have been highlighted with circled numbers. For instance, as observed in number 1 and 2, the status of the current context and the status of the current available modalities are displayed all the time. This design decision takes into account the guideline ‘*Provide modalities and context status feedback*’.

The first window displays a calendar view component as well as a list view that displays the recently created events (number 3). Also using the left(<<) and right arrow (>>) situated next to the month title, it is possible to move between the calendar months. Then, for the context of this window the motion gestures *left* and *right* produce the same result. The second window is displayed when a user creates an event for the current day, either by using the “shake” gesture, pressing the “new event” button or by approaching a NFC tag to the device.

The window shown on the left hand side of Figure 4.7 displays a form with the basic fields to create a calendar event. Inside this window, it is important to highlight the area marked with the number 4. Within this area, users are able to draw the *checkmark* or *line* symbols to save or cancel the creation of the event. This area only appears when the symbol-drawing interaction is allowed. Likewise, in this context the “shake” gesture executes a save action and the “face down” gesture a cancel action.

Finally, the date picker component that is displayed when a user wants to change the starting date of the event, allows to move forward and back the month’s

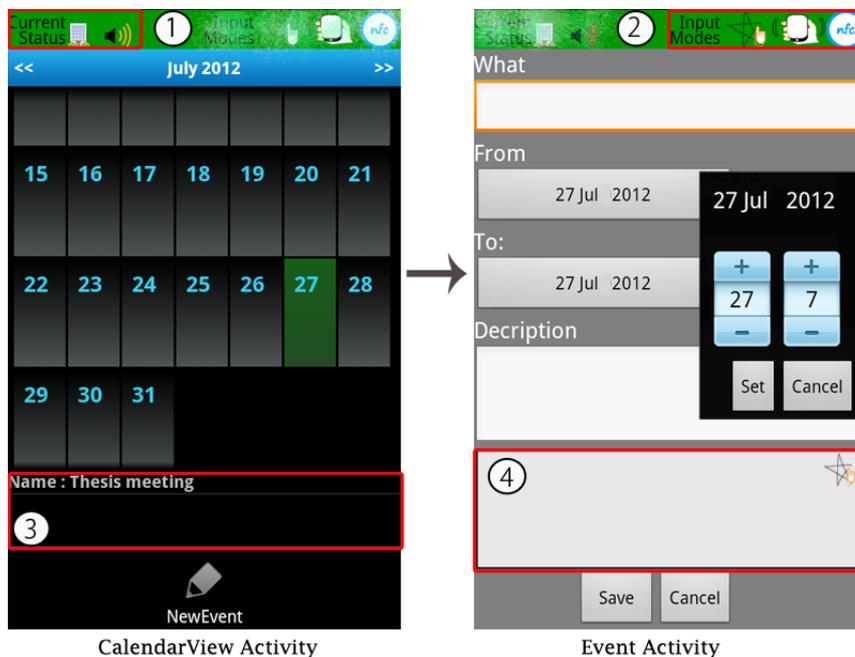


Figure 4.7: User interface

days using two modalities. A day is increased or decreased either using the traditional single tap interaction or using motion gestures (tilting back and forward the device).

4.5.2 Recognition of Input Modalities

The *Activities* are in charge of recognising the different events resulting from the inputs performed by the users. Since each modality is associated to different Android event classes, the `EventOfInterest` class was defined to refer from one single class to all these events.

Hence, indistinctly of the type of event captured by the activities, the `MultimodalController` receives an event of the type `EventOfInterest`. Important fields of this class are *uicontext* and *type*. The *uicontext* refers to the Activity that is running in the foreground when the input modality is executed. The *type* allows to specify which specific interaction technique was used by the user such as *draw check mark* or *draw line*.

Therefore, a set of enumerated types to define the possible types of interactions techniques for each modality were defined (`NfcInteraction`, `AccInterac-`

tion, FingerGesturesInteraction). Likewise, the possible values that the UI context can take were defined in the UIcontext enumeration.

Additionally, three classes corresponding to the supported input modalities extend from the EventOfInterest class, namely the AccelerometerEvent, NfcEvent and TouchEvent. The main methods and attributes are shown in Figure 4.8 and explained below.

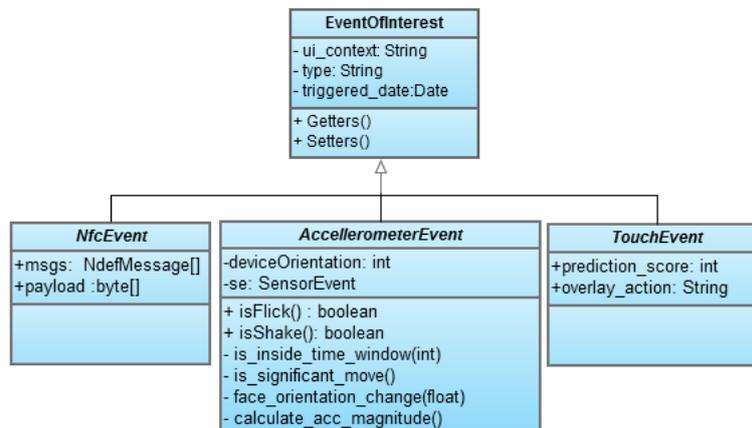


Figure 4.8: EventOfInterest class and subtypes

NfcEvent

As previously described, one of the supported input modalities in MAA is the use of extra gestures (tangible interaction). This type of user interaction can be implemented using RFID technology. For the development of this application, it was decided to use Near Field Technology (NFC) to take advantage of the built-in readers that come along with the modern smartphones.

Specifically, the Samsung Google Nexus S was used to test and run the application. This smartphone is an NFC-enabled Android phone that allows to read/write NDEF formatted information. The type of NFC tags that were used are Mifare classic tags, specifically the Trikker-1k CT50⁵. The left-hand side of Figure 4.9 illustrates the intended affordance of NFC tags within the MAA context. On the right-hand side the same concept is illustrated using a real smartphone device and a NFC Tag.

⁵http://www.nfcnetstore.com/pdf/Trikker-1k_CT50.pdf

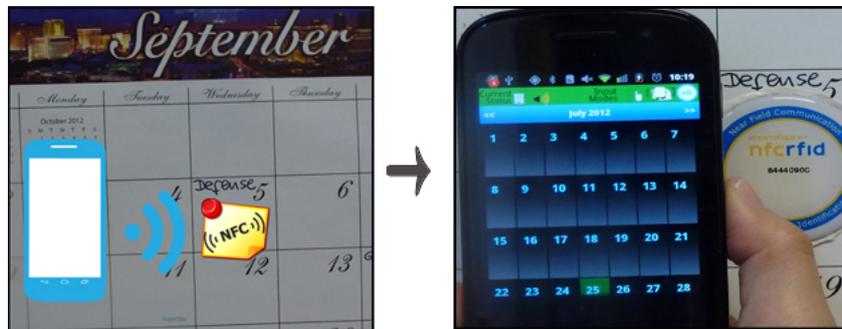


Figure 4.9: NFC calendar events

The specific interaction that the application supports is to create a new event by touching a NFC tag.

The Android SDK has a special package (`android.nfc`) that provides access to the Near Field Communication (NFC) functionality. Using these classes, the `BaseActivity` is responsible to access the device's NFC hardware and properly handle new NFC intents. When a NFC tag is discovered, the corresponding `NfcEvent` is created and dispatched to the `MultimodalController`. The fields `msgs` and `payload` refer to the content that the tag can store or read from the `nfc` intent.

AccelerometerEvent

The `AccelerometerEvent` class represents the events produced by the gestures executed in thin air with the device. These gestures are captured by the accelerometer sensor. As previously described, the gestures supported by the application are navigation gestures (flick left, right, backwards and forwards), the *face down* and *shake* gesture.

The `BaseActivity` is in charge of accessing the device's sensors and listens to the variations in the accelerometer data. However, the Android SDK does not provide methods that recognise specific gestures. Therefore, the `AccelerometerEvent` class is in charge of this task. This class provides methods to recognise two types of motion gestures, namely *flick* and *shake* gestures. Moreover, the `isFlick` method returns the specific direction of the flick movement.

The main idea behind the recognition of flick gestures relies on the analysis of the acceleration values on the x,y,z axis. Using an accelerometer monitor application, a pattern was noticed when executing the flick movements. As observed in Figure 4.10, when performing the *left* and *right* flick, changes in the the sign of the x axis acceleration values are observed. In turn, as observed in Figure 4.11 when executing the *backward* and *forward* flick, changes in the sign of the y axis acceleration values were noticed. The same behaviour was observed, when putting the device face down. The acceleration in the z axis resulted always negative. This observation is the main criteria to classify the data received from the accelerometer sensor. However, due to the extreme sensitivity of the accelerometer sensor, three additional considerations were taken into account for the gesture's recognition. A brief explanation of each consideration is described below.

- ▷ **Define a time window:** A one second time window was established to analyse the accelerometer readings inside this interval. Otherwise, when executing one gesture with the device, more than one recognition resulted true and multiple `AccelerometerEvent` objects were instantiated.
- ▷ **Identify significant movements:** It was important to analyse if the movement performed by the user was considered significant or not. Therefore, we analysed the standard deviation of the magnitude of the acceleration readings within the time window. Based on the experimental results found in [106] and the analysis we performed using the stored accelerometer readings when a person was walking, we established a movement threshold value of 0.7. A movement that exceeds this threshold, was considered as a significant move.
- ▷ **Significant variations on each axis acceleration:** We evaluated the difference between the readings from the current event acceleration data and the data from the last stored reading. A threshold which exceeds 0, represents an attempt to detect significant left-right or up-down moves.

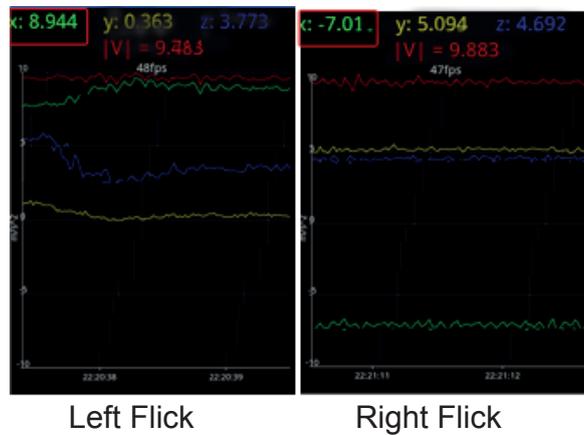


Figure 4.10: Acceleration readings while executing *left* and *right* flick gestures

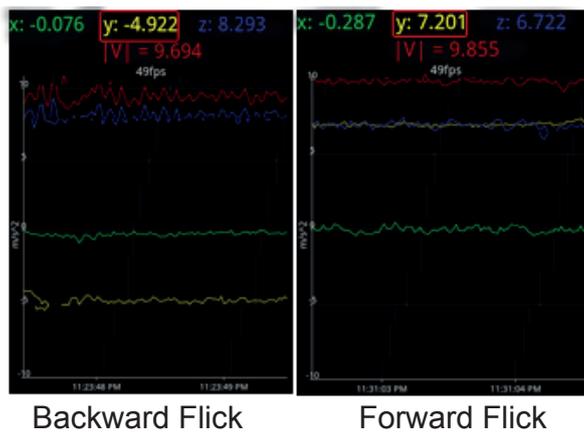


Figure 4.11: Acceleration readings while executing *back* and *forward* flick gestures

Finally, a slightly different approach was used to recognise the *shake* gesture. As observed in Figure 4.12, when the shake gesture was executed, the motion is continuous in a time interval and several changes in the acceleration direction are observed. For the recognition of this gesture, the rate of change of acceleration with respect to time (the Jolt⁶) was calculated to check the smoothness of the motion. If the calculated rate exceeds a predefined force threshold, then the duration of the shake and the number of shakes were evaluated. These three thresholds

⁶[http://en.wikipedia.org/wiki/Jerk_\(physics\)](http://en.wikipedia.org/wiki/Jerk_(physics))

were established experimentally and supported by the information provided by an Android jerk-meter application.

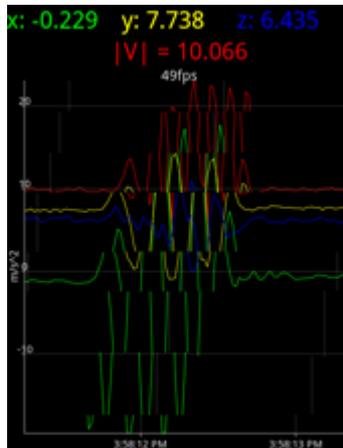


Figure 4.12: Acceleration readings when executing the *shake* gesture

Touch Event

The `TouchEvent` class represents the symbol gestures executed on the touch screen with the fingers. Figure 4.13 illustrates the supported gestures in the application. The *check mark* gesture executes a save action, whereas the *line* gesture executes a cancel action. The recognition of these gestures relies in the gesture recognition toolkit provided with the Android SDK. It is worth to notice that this toolkit implements the Protractor algorithm [63], which was proven to performed better than the \$1 gesture recognizer[118].



Figure 4.13: Recognised gestures

Using the *GesturesBuilder* tool provided by the Android SDK, it was possible to create a custom gestures library. The library provides a method to evaluate a

user's gesture against the gestures stored in the library. This method returns a prediction score. Although in most of the examples a value of 1.0 represents a good match, based on experimental observations we decided to establish a prediction threshold of 5.0 to reduce the number of false recognitions.

4.5.3 The Multimodal Controller and Fusion Manager

Figure 4.14 shows the `MultimodalController` class and related classes. It can be observed that this class is in charge of two main functions:

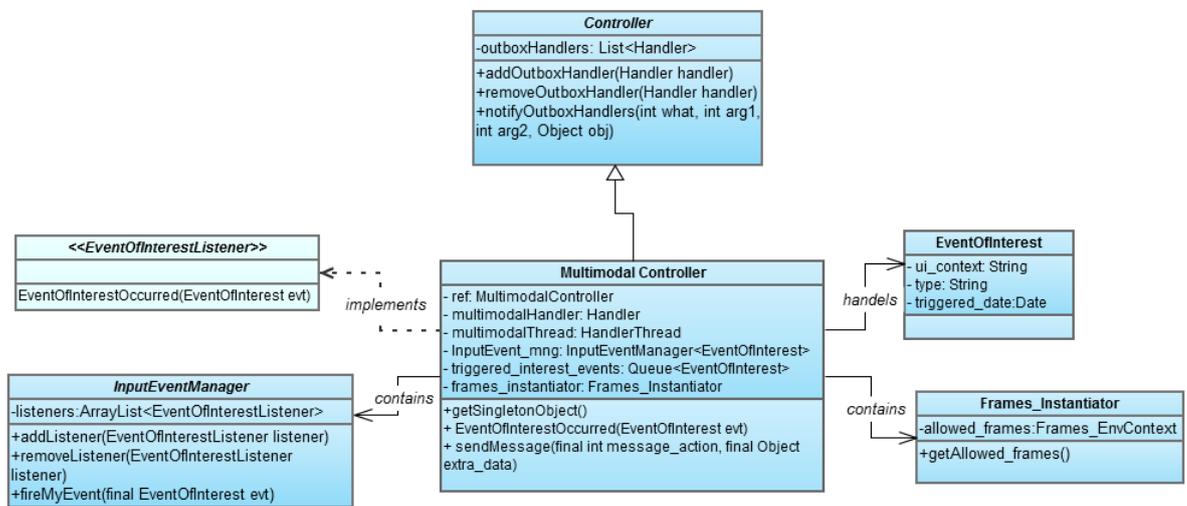


Figure 4.14: The `MultimodalController`

- ▷ Listen and handle `EventOfInterest` events. Specifically, the `InputEventManager` allows to register event listeners and publish events from the Activities.
- ▷ Send UI-action messages to the different Activities that are registered by the `MultimodalController` as external Handlers.

Thus, handling `EventOfInterest` events refers to the evaluation of the incoming events by the *Fusion Manager* component. Second, after receiving a result from the *Fusion Manager*, the `MultimodalController` is in charge of sending a message to the `BaseActivity` (`DialogueManager`). The message contains the action that must be executed in the corresponding view. Some specific considerations had to be taken into account to achieve the communication

between the `BaseActivity` and the `MultimodalController`.

First, the `MultimodalController` was defined as a singleton class to prevent the direct instantiation from other classes. Second, all the multimodal tasks are processed by a separate thread to avoid blocking the user interface. Associated to this thread is a `MultimodalHandler`. This handler allows to send and process `Runnable` objects associated with the thread's message queue. In this way, it is possible to isolate the multimodal processes from the processes executed in the interface (main thread). Moreover, it is possible to pass data back and forth between the main thread and the multimodal thread.

To sum up, when the Fusion Manager component resolves which action has to be sent to the UI, the `MultimodalHandler` posts a `Runnable` that contains a call to `notifyOutboxHandler()` method. This method sends messages to the specific registered external Handlers (Activities).

The Fusion Manager

The Fusion Manager component encompasses three main classes: `FrameEnvContext`, `Frame` and `Slot`. Figure 4.15 illustrates the relationship between the classes and their main attributes.

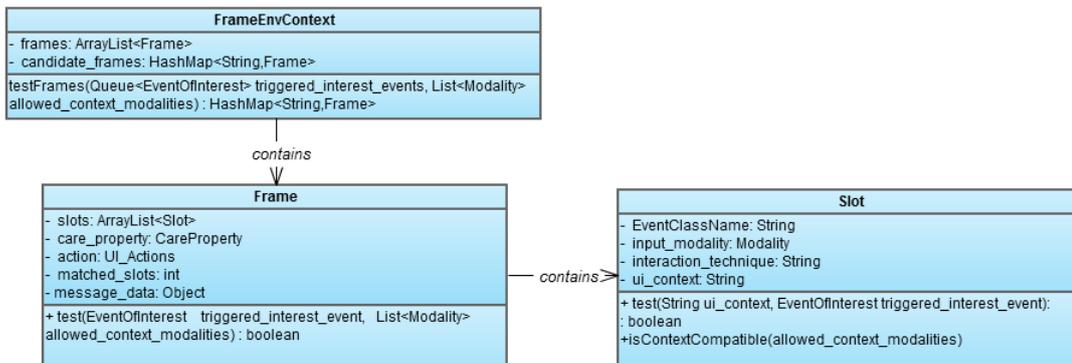


Figure 4.15: Fusion Manager classes

As reviewed in the Multimodal Interaction background section, the fusion engine is in charge of capturing the different events and provides an interpretation of the user intent. To achieve this, MAA performs fusion at the decision level and uses a frame-based approach. As stated by Vo and Wood [42], frame-based fusion

uses data structures called frames and slots to represent the data coming from the multiple sources or input modalities.

As observed in Figure 4.14, the `Frame` class contains the information about the UI-action that should be executed when given modalities are performed. Each *Frame* can have one or multiple *Slots* and the *care property* field specifies which type of combination between the modalities is allowed. On the other hand, the slot stores the information about the interaction technique and input modality such as *shake* and *motion gestures*. Furthermore, the *Slot* specifies in which Activity and UI context, the interaction is permitted.

Each class provides a test method that evaluates the incoming events at different levels. To sum up the fusion process, the evaluation method from the class `FrameEnvContext` receives the `EventOfInterest`'s `Queue` associated to the `MultimodalController` as well as the set of allowed input modalities. At this point, each queued event is evaluated against all the frames that were instantiated by the application. Six frames were statically defined and instantiated by the `FramesInstantiator` class. These frames represent the possible interaction techniques defined in Section 4.2.2.

The evaluation at the frame level ensures that all the slots associated to it are evaluated. However, it is worth to notice that only the slots that are context compatible are evaluated at the slot level. This means that first they verified if the slots are compatible with the supported modalities provided by the *PolicyManager* component.

For instance, as it can be observed in Figure 4.16, two slots (S1 and S2) are defined for the frame F1. However, only S2 will be further evaluated. The slot S1 is discarded at the frame level since motion gestures are not listed as an allowed modality.

Finally, the evaluation at the slot level verifies if the the interaction technique, modality and UI context from the triggered event match with the slot information.

To better illustrate the context fusion algorithm, Figure 4.17 and Figure 4.18 display the Android console results when the motion gesture *shake* was performed with different context conditions.

As one can notice in Figure 4.17, a *shake* gesture is evaluated by the fusion engine. However, no match was found for it since the context conditions only allowed to interact with tangible interaction. It is really important to notice that

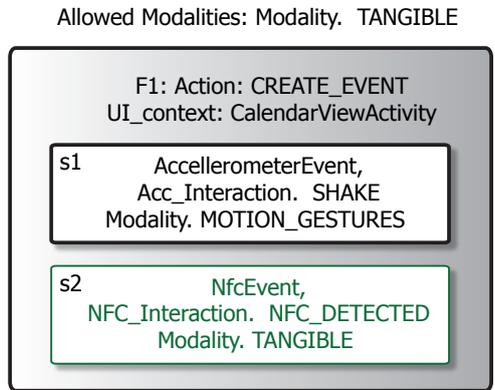


Figure 4.16: Context frame

only the Frame_0 was evaluated at the slot level, since one of his slots contain the supported modality. The slots from the other five frames were discarded.

P.	A.	Tag	Text
23:17:29.392	2	c. MultimodalEventsC...	Event to add in the Queue; AccelerometerEvent , CalendarViewActivity , Shake
23:17:29.392	2	c. MultimodalEventsC...	Allowed ModalityTANGIBLE

Time	P.	A.	Tag	Text
08-01 23:17:29.392	2	c.	Frame	Slot:..NfcEvent,NFC_DETECTED
08-01 23:17:29.392	2	c.	Frames_EnvContext	Frame:,Frame_0 matched-slots:0
08-01 23:17:29.392	2	c.	Frames_EnvContext	Frame:,Frame_1 matched-slots:0
08-01 23:17:29.392	2	c.	Frames_EnvContext	Frame:,Frame_2 matched-slots:0
08-01 23:17:29.392	2	c.	Frames_EnvContext	Frame:,Frame_3 matched-slots:0
08-01 23:17:29.392	2	c.	Frames_EnvContext	Frame:,Frame_4 matched-slots:0
08-01 23:17:29.392	2	c.	Frames_EnvContext	Frame:,Frame_5 matched-slots:0

Figure 4.17: No matching slot

When the context conditions changed and the extra gestures (tangible) and motion gestures were set as the allowed modalities, we can see in Figure 4.18 that all the slots containing the supported modality where evaluated. However, only for the Frame_0 a match was found. It is also important to notice that Frame_4 and Frame_5 do not display the slots associated with finger touch gestures. The modality is not permitted for the context conditions.

	P.	A.	Tag	Text	
23:29:28.966	3	c.	MultimodalEventsC...	Event to add in the Queue; AccelerometerEvent , CalendarViewActivity , Shake	
23:29:28.966	3	c.	MultimodalEventsC...	Allowed ModalityTANGIBLE	
23:29:28.966	3	c.	MultimodalEventsC...	Allowed ModalityMOTION_GESTURES	
08-01	23:29:28.970	3	c.	Frame	Slot:..NfcEvent,NFC_DETECTED
08-01	23:29:28.981	3	c.	Frame	Slot:..AccelerometerEvent,Shake
08-01	23:29:28.985	3	c.	Frame	Slot: (match)..AccelerometerEvent,Shake
08-01	23:29:28.985	3	c.	Frames_EnvContext	Frame:.,Frame_0 matched-slots:1
08-01	23:29:28.989	3	c.	Frame	Slot:..AccelerometerEvent,Right
08-01	23:29:29.001	3	c.	Frames_EnvContext	Frame:.,Frame_1 matched-slots:0
08-01	23:29:29.001	3	c.	Frame	Slot:..AccelerometerEvent,Left
08-01	23:29:29.013	3	c.	Frames_EnvContext	Frame:.,Frame_2 matched-slots:0
08-01	23:29:29.013	3	c.	Frame	Slot:..AccelerometerEvent,Down
08-01	23:29:29.024	3	c.	Frames_EnvContext	Frame:.,Frame_3 matched-slots:0
08-01	23:29:29.024	3	c.	Frame	Slot:..AccelerometerEvent,Shake
08-01	23:29:29.024	3	c.	Frames_EnvContext	Frame:.,Frame_4 matched-slots:0
08-01	23:29:29.024	3	c.	Frame	Slot:..AccelerometerEvent,FaceDown
08-01	23:29:29.024	3	c.	Frames_EnvContext	Frame:.,Frame_5 matched-slots:0

Figure 4.18: Slot match

4.5.4 The Context Controller and Policy Manager

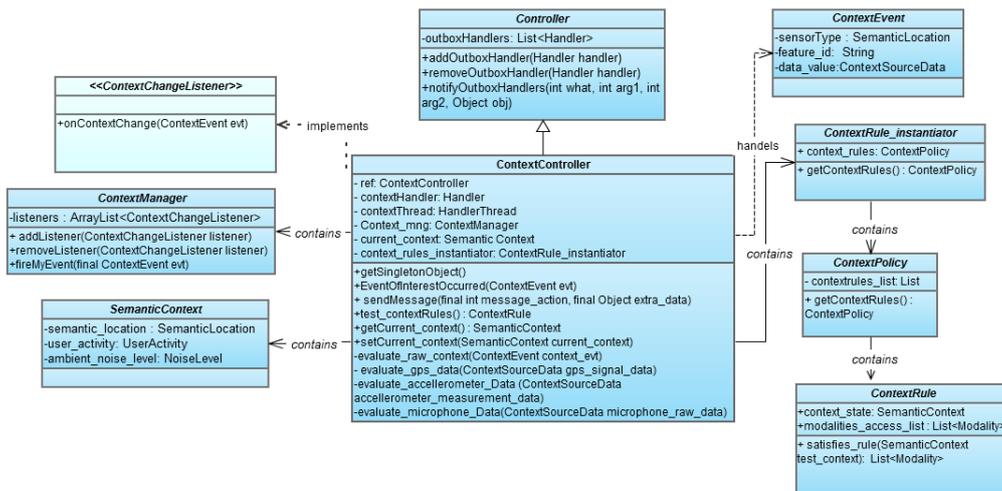


Figure 4.19: ContextController

The ContextController class along with the PolicyManager component constitutes the adaptive mechanism for the application. Figure 4.17 illustrates the most relevant classes associated to these components. The system-induced adaptation process described in [84] was taken into account for the implementation of these components.

The `ContextController` is in charge of capturing and interpreting all the adaptation triggers received from the different background services. To be consistent with the adaptation triggers that were defined during the design phase, three background services are constantly monitoring the environment activity, namely (`NoiseLevelService`, `MovementDetectionService`, `GPSService`).

Each service has specific thresholds defined that trigger a possible adaptation call. The adaptation call is sent to the `MultimodalController` in the form of a `ContextEvent`. This type of event carries the raw information captured by the different sensors. To deal with the different types of data obtained from the sensors (decibels, acceleration, GPS coordinates), the `ContextEvent` carries this information as a `ContextSourceData` object.

The information that is carried by the `ContextEvent` does not possess any semantic meaning for the application. For instance, the `ContextEvent` resulting from a change in the noise level readings contains the `spl` numeric value (sound pressure level). Hence, the `MultimodalController` is in charge of interpreting this input data and then update the *current context* model. In this way, the `spl` numeric value is evaluated and categorised as low, medium or high.

Based on the up-to-date information from the *current context* model, the system decides upon the necessity of an adaptation. The decision making process is responsible of the *PolicyManager* component.

Policy Manager

The Policy Manager consists of two classes: `ContextPolicy` and `ContextRule` classes.

The adaptation policy approach that MAA follows is rule-based. Twelve context rules were statically defined and instantiated by the `ContextRuleInstantiator`. The data structures `ContextPolicy` and `ContextRule` are used to represent the rules defined in Section 4.2.3. Thus, the `ContextPolicy` class represents a set of context rules and a `ContextRule` is defined by a context state and a set of allowed modalities. A context state specifies information related to the location, user activity and noise level.

In this way, every time an adaptation call is received, the current context is evaluated against all the established rules. In MAA, a rule is satisfied if all the parameters from the current context model (semantic location, user activity, noise level) match with the context parameters from one of the established rules. Then,

the modalities associated with the matched rule are said to be the suitable modalities for that context.

The last step of the adaptation mechanism covers the process of making the adaptation visible to the user. To achieve this, the `ContextController` notifies the `DialogueManager (BaseActivity)` when the `ContextPolicy` component provides an answer. The same message-based communication described in the `MultimodalController` section is used here. However, the type of actions that the `ContextController` sends to the interface are fixed to *update the context status* and *update the input modes*. In this way, the changes are reflected to the user as shown in Figure 4.20 and Figure 4.21.

Figure 4.20 displays the suitable modalities for the semantic location *indoor* and different noise level variations. However, as can be seen in the image, when the systems detects that the user is walking, it is not possible to interact using the motion gestures modality.

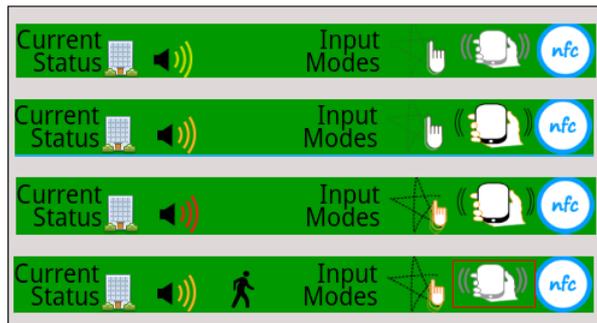


Figure 4.20: Suitable modalities for the *indoor* location and different *noise level* values

Similarly, Figure 4.21 shows the effect of an adaptation when the location of the user is set as *outdoors*. As observed in the image, the available modalities under the same noise level are different in comparison to the modalities observed in Figure 4.20 and defined for the *indoors* location.

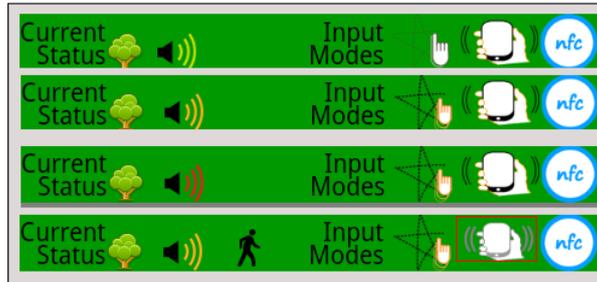


Figure 4.21: Suitable modalities for *outdoors* location and different *noise level* values

4.5.5 Summary

The current chapter described design and technical aspects for the development of the adaptive multimodal mobile agenda application (MAA).

We have shown how the proposed guidelines were used as a basis to design the application. Furthermore, it is explained how specific components from the application select and allow to visualise the most appropriate modalities according to context conditions.

Finally, the description of conceptual models that represent the input modalities, the interaction techniques as well as the context rule are explained in detail.

Chapter 5

Conclusions and Future Work

5.1 Summary

As described in the introduction of this thesis, the expected outcomes of this work were a systematic investigation of the field of mobile multimodal interaction and a proof of concept application based on the study findings and guidelines.

We have first conducted a deep study of the relevant fields related to this thesis, including multimodal interaction, mobile interaction, context awareness and adaptive interfaces.

Then, we have selected, classified and analysed relevant research projects based on a set of classification parameters and selection criteria. The result of the classification of research articles is showcased in a number of recapitulative tables. Based on the literature review and the article classification, we have analysed and discussed three aspects, namely the observed modality combinations, the influence of the context in the usage of different input modalities and system-induced adaptation core features. As a result of these analyses, we proposed a set of guidelines to facilitate the design process of a context-aware adaptive multimodal mobile application.

Finally, based on the study findings we presented the Multimodal Adaptive Agenda (MAA) as a proof of concept implementation. We considered three main aspects during the design and development phase of the application, namely the use of the proposed design guidelines, the use of the less explored modalities and the combination of modalities and the communication between the fusion engine and the adaptation mechanism.

This thesis makes the following three main contributions:

The first major contribution is a systematic study of context-dependent adaptation in mobile multimodal interfaces that permits new researchers to compare what has been done and can be done in the field. To sum up, the study demonstrates that the field of system-induced adaptation has been barely addressed. Moreover, it indicates that some modalities, such as motion gestures and extra gestures, have been less explored although they have received good acceptance rates in different user study evaluations. Finally, our investigation provides some insights on the suitability of modalities based on physical conditions, social conditions and location.

The second contribution is a set of guidelines which can be used as a starting point by future designers or developers during the design phase of this type of adaptive multimodal applications. These guidelines summarise the core features observed in the reviewed papers as well as already existing guidelines from the related fields.

Finally, the third contribution is a proof of concept application that showcases the technical feasibility of integrating adaptation and the multimodal functionality offered by modern smartphones. Moreover, an additional contribution is the exploration of a fusion algorithm that takes context parameters as an additional input when interpreting user input events.

5.2 Future Work

Due to the limited time, we focussed on the exploration and analysis of input channel adaptation influenced by environmental factors. However, it would be interesting to extend the analysis and include adaptation influenced by the user and device capabilities. In this way, the analysis of all the elements that constitute the context of interaction of an user could be achieved.

In regard to the Adaptive Multimodal Agenda application, it would be interesting to conduct a field study with users to evaluate the usability and effectiveness of system-induced adaptation in conjunction with multimodal interaction. To conduct the study in good conditions, some aspects should be taken into account and enhanced in the application. Currently, the application presents some basic functionality to showcase the underlying concept. However, more functionality associated to the extra gestures modality should be included. For instance, the capability of reading a NFC calendar event or exporting a calendar event from

the application to a NFC tag would drastically improve the usability. Likewise, the finger-based recognition can be enhanced to support more symbols, particularly numbers that could be used to change the day or month in the calendar picker.

Furthermore, the application currently relies on a basic motion gesture recognition mechanism. However, better recognition rates and more complex gestures could be included by using machine learning-based recognition classifiers. Likewise, the toolkit provided by Android provides a quick and good workaround for finger-based symbol recognition, but it would be interesting to extend this functionality with a gesture prediction display, where a set of gestures with similar prediction scores are shown to the user. This functionality could help to reduce the number of misrecognised gestures in the case of similar gestures, such as the *checkmark* and *cross* symbols. Finally, output feedback could be incorporated to reduce the necessary attention when performing a gesture even more.

From a development perspective, the fusion algorithm could be enhanced to support redundant and complementary composition of modalities. Additionally, the description of the allowed modalities and interaction techniques should rely on a XML-based language such as SMUIML [32]. Likewise, the definition of the context-multimodal rules should be formalised and described using an XML-based language too. Based on this seminal work, a rapid prototyping tool for the development of context-aware mobile multimodal applications could be further developed. Such a tool or framework would significantly simplify and speed up the development of future adaptive mobile multimodal applications.

Bibliography

- [1] Near Field Communication Versus Bluetooth. "<http://www.nearfieldcommunicationnfc.net/nfc-vs-bluetooth.html>".
- [2] Seven Usability Guidelines for Websites on Mobile Devices. "{<http://www.webcredible.co.uk/user-friendly-resources/web-usability/mobile-guidelines.shtml>}", year = 2007.
- [3] Understanding NFC Data Exchange Format (NDEF) Messages - Nokia Developer Wiki. "[http://www.developer.nokia.com/Community/Wiki/Understanding_NFC_Data_Exchange_Format_\(NDEF\)_messages](http://www.developer.nokia.com/Community/Wiki/Understanding_NFC_Data_Exchange_Format_(NDEF)_messages)".
- [4] Worldwide Mobile Device Sales to End Users by Vendor in 3Q11. "<http://www.gartner.com/it/page.jsp?id=1848514>".
- [5] Worldwide mobile telephone terminal sales estimates for 1998. "http://www.gartner.com/5_about/press_room/pr19990208a.html".
- [6] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a Better Understanding of Context and Context-Awareness. In *Proceedings of HUC 1999, 1st International Symposium on Handheld and Ubiquitous Computing*, pages 304–307, Karlsruhe, Germany, September 1999.
- [7] A. Andersen and R. Karlson. Experimenting with Instant Services Using NFC Technology. In *Proceedings of SMART 2012, 1st International Conference on Smart Systems, Devices and Technologies*, pages 73–78, Stuttgart, Germany, May 2012.
- [8] C. Ardito, P. Buono, M. F. Costabile, R. Lanzilotti, and A. Piccinno. A Tool for Wizard of Oz Studies of Multimodal Mobile Systems. In *Proceedings of*

- HSI 2009, 2nd Conference on Human System Interactions*, pages 341–344, Catania, Italy, 2009.
- [9] P. Atrey, M. Hossain, A. El Saddik, and M. Kankanhalli. Multimodal Fusion for Multimedia Analysis: A Survey. *Multimedia Systems*, 16(6):345–379, 2010.
- [10] N. Z. b. Ayob, A. R. C. Hussin, and H. M. Dahlan. Three Layers Design Guideline for Mobile Application. In *Proceedings of the ICIME 2009, International Conference on Information Management and Engineering*, pages 427–431, April 2009.
- [11] M. Beaudouin-Lafon. Designing Interaction, Not Interfaces. In *Proceedings of AVI2004, International Working Conference on Advanced Visual Interfaces*, pages 15–22, Gallipoli, Italy, 2004.
- [12] M. Bezold and W. Minker. *Adaptive Multimodal Interactive Systems*. Springer Publishing Company, Incorporated, 1st edition, 2011.
- [13] R. A. Bolt. “Put-That-There”: Voice and Gesture at the Graphics Interface. In *Proceedings of ACM SIGGRAPH 1980, 7th International Conference on Computer Graphics and Interactive Techniques*, pages 262–270, Seattle, USA, July 1980.
- [14] D. Bühler, W. Minker, J. Häussler, and S. Krüger. The SmartKom Mobile Multi-Modal Dialogue System. In *Proceedings of ISCA Tutorial and Research Workshop (ITRW) on Multi-Modal Dialogue in Mobile Environments*, Irsee, Germany, June 2002.
- [15] H. Bunt. Issues in Multimodal Human-Computer Communication. In *Multimodal Human-Computer Communication*, pages 1–12. Springer Berlin / Heidelberg, 1998.
- [16] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt. A Unifying Reference Framework for Multi-Target User Interfaces. *Interacting with Computers*, 15(3):289–308, 2003.
- [17] A. Celentano and O. Gaggi. Context-Aware Design of Adaptable Multimodal Documents. *Multimedia Tools and Applications*, 29(1):7–28, April 2006.
- [18] G. Chen and D. Kotz. A Survey of Context-Aware Mobile Computing Research. 2000.

- [19] L. Chittaro. Distinctive Aspects of Mobile Interaction and Their Implications for the Design of Multimodal Interfaces. *Journal on Multimodal User Interfaces*, 3:157–165, April 2010.
- [20] D. Costa and C. Duarte. Adapting Multimodal Fission to User’s Abilities. In *Proceedings of UAHCI 2011, 6th International Conference on Universal access in Human-Computer Interaction: Design for all and eInclusion - Volume Part I*, pages 347–356, Orlando, USA, 2011.
- [21] J. Coutaz, L. Nigay, D. Salber, A. Blandford, J. May, and R. M. Young. Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The CARE Properties. In *Proceedings of INTERACT 1995, 5th International Conference on Human-Computer Interaction*, Lillehammer, Norway, June 1995.
- [22] F. Cutugno, V. A. Leano, R. Rinaldi, and G. Mignini. Multimodal Framework for Mobile Interaction. In *Proceedings of the AVI 2012, International Working Conference on Advanced Visual Interfaces*, pages 197–203, Capri Island, Italy, 2012.
- [23] R. Dale, H. Moisl, and H. Somers, editors. *The Generation of Multimedia Documents. In: A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*. Marcel Dekker Inc., 2000.
- [24] M. Dalmau, P. Roose, and S. Laplace. Context Aware Adaptable Applications, A Global Approach. *International Journal of Computer Science Issues*, 1:pp13–25, 2009.
- [25] L. David, M. Endler, S. D. J. Barbosa, and J. V. Filho. Middleware Support for Context-Aware Mobile Applications with Adaptive Multimodal User Interfaces. In *Proceedings of U-Media 2011, 4th International Conference on Ubi-Media Computing*, pages 106–111, São Paulo, Brazil, July 2011.
- [26] M. de S, C. Duarte, L. Carriço, and T. Reis. Designing Mobile Multimodal Applications. In *Multimodality in Mobile Computing and Mobile Devices: Methods for Adaptable Usability*, pages 106–136. Information Science Reference - Imprint of: IGI Publishing, 1st edition, 2009.
- [27] M. de Sá and L. Carriço. Lessons From Early Stages Design of Mobile Applications. In *Proceedings of MobileHCI 2008, 10th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 127–136, Amsterdam, The Netherlands, 2008.

- [28] A. K. Dey and J. Häkkinen. Context-Awareness and Mobile Devices. In *Context-Awareness and Mobile Devices. In Handbook of Research on User Interface Design and Evaluation for Mobile Technology*, chapter XIII, pages 205–217. 2008.
- [29] J. Doyle, M. Bertolotto, and D. Wilson. A Survey of Multimodal Interfaces for Mobile Mapping Applications. In L. Meng, A. Zipf, and S. Winter, editors, *Map-based Mobile Services*, Lecture Notes in Geoinformation and Cartography, pages 146–167. Springer Berlin Heidelberg, 2008.
- [30] C. Duarte and L. Carriço. A Conceptual Framework for Developing Adaptive Multimodal Applications. In *Proceedings of IUI 2006, 11th International Conference on Intelligent User Interfaces*, pages 132–139, Sydney, Australia, 2006.
- [31] A. D’Ulizia. Exploring Multimodal Input Fusion Strategies. In *Multimodal Human Computer Interaction and Pervasive Services*, pages 34–57. IGI Publishing, 2009.
- [32] B. Dumas, D. Lalanne, and S. Oviatt. Human Machine Interaction. chapter Multimodal Interfaces: A Survey of Principles, Models and Frameworks, pages 3–26. Springer-Verlag, Berlin, Heidelberg, 2009.
- [33] T. Dutoit and S. Dupont. *Multimodal Signal Processing - Theory and Applications for Human-Computer Interaction*. Academic Press, 2010.
- [34] C. Efstratiou. *Coordinated Adaptation for Adaptive Context-aware Applications*. Ph.d. thesis, Lancaster University, Computing Department, 2004.
- [35] M. El Choubassi, O. Nestares, Y. Wu, I. Kozintsev, and H. Haussecker. An Augmented Reality Tourist Guide on Your Mobile Devices. In *Proceedings of MMM 2010, 16th International Conference on Advances in Multimedia Modeling*, pages 588–602, Chongqing, China, 2010.
- [36] L. Esmahi and E. Badidi. An agent-based framework for adaptive M-learning. In *Proceedings of IEA/AIE 2004, 17th International Conference on Innovations in Applied Artificial Intelligence*, pages 749–758, Ottawa, Canada, May 2004.
- [37] G. Evans and D. Johnson. Stress and Open-Office Noise. *Journal of Applied Psychology; Journal of Applied Psychology*, 85(5):779, 2000.
- [38] M. E. Foster. State-of-the-art Review: Multimodal Fission. *COMIC project Deliverable*, 6(09), 2002.

- [39] K. Z. Gajos, M. Czerwinski, D. S. Tan, and D. S. Weld. Exploring the Design Space for Adaptive Graphical User Interfaces. In *Proceedings of AVI 2006, the Working Conference on Advanced Visual Interfaces*, pages 201–208, Venezia, Italy, 2006.
- [40] M. Gargenta. *Learning Android*. O’Reilly Media, Inc., 2011.
- [41] D. Gentner and J. Nielsen. The Anti-Mac Interface. *Commun. ACM*, 39(8):70–82, August 1996.
- [42] R. A. Goubran and C. Wood. Building an Application Framework for Speech and Pen Input Integration in Multimodal Learning Interfaces. In *Proceedings of the ICASSP 1996, IEEE International Conference*.
- [43] P. Grifoni. Multimodal Fission. In *Multimodal Human Computer Interaction and Pervasive Services*, pages 103–120. IGI Publishing, 2009.
- [44] N. Henze, E. Rukzio, and S. Boll. 100,000,000 Taps: Analysis and Improvement of Touch Performance in the Large. In *Proceedings of Mobile-HCI 2011, 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 133–142, Stockholm, Sweden, 2011.
- [45] R. Hill and J. Wesson. A-POInter: An Adaptive Mobile Tourist Guide. In *Proceedings of SAICSIT 2010, Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*, pages 113–122, Bela Bela, South Africa, October 2010.
- [46] H. Ishii and B. Ullmer. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of CHI 1997, 15th International Conference on Human Factors in Computing Systems*, pages 234–241, Atlanta, USA, March 1997.
- [47] R. Jacob, A. Girouard, L. M. Hirshfield, M. Horn, O. Shaer, E. T. Solovey, and J. Zigelbaum. What is the Next Generation of Human-Computer Interaction? *ACM Interactions*, 14(3):53–58, May 2007.
- [48] R. J. Jacob, A. Girouard, L. M. Hirshfield, M. S. Horn, O. Shaer, E. T. Solovey, and J. Zigelbaum. Reality-based Interaction: A Framework for Post-WIMP Interfaces. In *Proceedings of CHI 2008, 26th Annual SIGCHI Conference on Human Factors in Computing Systems*, pages 201–210, Florence, Italy, April 2008.

- [49] A. Jaimes and N. Sebe. Multimodal Human-Computer Interaction: A Survey. *Computer Vision and Image Understanding*, 108(1–2):116–134, October 2007.
- [50] S. Jain. Introduction to Mobile Computing. *Crossroads*, 7(2):2–, December 2000.
- [51] M. Johnston, P. R. Cohen, D. McGee, S. L. Oviatt, J. A. Pittman, and I. Smith. Unification-based multimodal integration. In *Proceedings of ACL 1998, 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 281–288, Madrid, Spain, 1997.
- [52] T. Jokela, N. Iivari, J. Matero, and M. Karukka. The Standard of User-Centered Design and the Standard Definition of Usability: Analyzing ISO 13407 Against ISO 9241-11. In *Proceedings of CLIHC 2003, the Latin American conference on Human-Computer Interaction*, pages 53–60, Rio de Janeiro, Brazil, 2003.
- [53] A. Joshi, S. Weerawarana, R. A. Weerasinghe, T. T. Drashansky, N. Ramakrishnan, and E. N. Houstis. A Survey of Mobile Computing Technologies and Applications. Technical report, 1995.
- [54] N. Kamel, S. A. Selouani, and H. Hamam. A Formal Approach to the Verification of Adaptability Properties for Mobile Multimodal User Interfaces. In *Multimodality in Mobile Computing and Mobile Devices: Methods for Adaptable Usability*, pages 53–74. Information Science Reference - Imprint of: IGI Publishing, 1st edition, 2009.
- [55] P. Klante, J. Krsche, and S. Boll. AccesSights A Multimodal Location-Aware Mobile Tourist Information System. In *Computers Helping People with Special Needs*, volume 3118 of *Lecture Notes in Computer Science*, pages 287–294. 2004.
- [56] J. Kong, W. Y. Zhang, N. Yu, and X. J. Xia. Design of Human-Centric Adaptive Multimodal Interfaces. *International Journal of Human-Computer Studies*, 69(12):854–869, December 2011.
- [57] A. Kratky. Gesture-Based User Interfaces for Public Spaces. In *Proceedings of UAHCI 2011, 6th International Conference on Universal Access in Human-Computer Interaction: Users Diversity - Volume Part II*, pages 564–572, Orlando, USA, 2011.

- [58] S. Kurkovsky. *Multimodality in Mobile Computing and Mobile Devices: Methods for Adaptable Usability*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA, 1st edition, 2009.
- [59] D. Lalanne, L. Nigay, P. Palanque, P. Robinson, J. Vanderdonckt, and J.-F. Ladry. Fusion Engines for Multimodal Input: A Survey. In *Proceedings of ICMI 2009, 11th International Conference on Multimodal Interfaces*, pages 153–160, Beijing, China, November 2009.
- [60] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell. A Survey of Mobile Phone Sensing. *Communications Magazine, IEEE*, 48(9):140–150, 2010.
- [61] T. Lavie and J. Meyer. Benefits and Costs of Adaptive User Interfaces. *International Journal of Human-Computer Studies*, 68(8):508–524, August 2010.
- [62] S. Lemmelä, A. Vetek, K. Mäkelä, and D. Trendafilov. Designing and Evaluating Multimodal Interaction for Mobile Contexts. In *Proceedings of ICMI 2008, 10th International Conference on Multimodal Interfaces*, pages 265–272, Chania, Greece, October 2008.
- [63] Y. Li. Protractor: A Fast and Accurate Gesture Recognizer. In *Proceedings of CHI 2010, 28th International Conference on Human Factors in Computing Systems*, pages 2169–2172, Atlanta, USA, April 2010.
- [64] V. López-Jaquero, J. Vanderdonckt, F. Montero, and P. González. Engineering interactive systems. chapter Towards an Extended Model of User Interface Adaptation: The Isatine Framework, pages 374–392. Springer-Verlag, 2008.
- [65] S. Love. *Understanding Mobile Human-Computer Interaction (Information Systems Series (ISS))*. Butterworth-Heinemann, Newton, MA, USA, 2005.
- [66] T. Lovett and E. O’Neill. *Mobile Context Awareness*. Springer Publishing Company, Incorporated, 2012.
- [67] U. Malinowski, K. Thomas, H. Dieterich, and M. Schneider-Hufschmidt. A Taxonomy of Adaptive User Interfaces. In *Proceedings of HCI 1992, Conference on People and Computers VII*, pages 391–414, York, United Kingdom, September 1992.

- [68] S. Mare, J. Sorber, M. Shin, C. Cornelius, and D. Kotz. Adapt-lite: Privacy-Aware, Secure, and Efficient mHealth Sensing. In *Proceedings of the WPES 2011, 10th Annual ACM Workshop on Privacy in the Electronic Society*, pages 137–142, Chicago, Illinois, USA, October 2011.
- [69] D. W. Mauney and C. Masterton. Small-Screen Interfaces. In *HCI Beyond the GUI: Design for Haptic, Speech, Olfactory, and Other Nontraditional Interfaces*, pages 307–354. Morgan Kaufmann Publishers Inc., 2008.
- [70] Z. Mednieks, L. Dornin, G. Meike, and M. Nakamura. *Programming Android*. O’Reilly Media, Inc., 2011.
- [71] P. Milgram and F. Kishino. A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information and Systems*, (12):1321, 1994.
- [72] N. Mitrovic and E. Mena. Adaptive User Interface for Mobile Devices. In *Proceedings of DSV-IS 2002, 9th International Workshop on Interactive Systems. Design, Specification, and Verification*, pages 29–43, Rostock, Germany, 2002.
- [73] J. Nielsen. Noncommand User Interfaces. *Communications of the ACM*, 36(4):83–99, April 1993.
- [74] L. Nigay and J. Coutaz. Multifeature Systems: The CARE Properties and Their Impact on Software Design. *Intelligence and multimodality in multimedia interfaces*, 1997.
- [75] A. Nijholt, D. Tan, B. Allison, J. del R. Milan, and B. Graitmann. Brain-Computer Interfaces for HCI and Games. In *Proceedings of CHI 2008, Extended Abstracts on Human Factors in Computing Systems*, pages 3925–3928, Florence, Italy, 2008.
- [76] R. Oppermann. *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software*. CRC, 1994.
- [77] R. Oppermann. Adaptively Supported Adaptability. *Int. J. Hum.-Comput. Stud.*, 40(3):455–472, March 1994.
- [78] R. Oppermann and R. Rasher. Adaptability and Adaptivity in Learning Systems. *Knowledge transfer*, 2:173–179, 1997.
- [79] S. Oviatt. Taming Recognition Errors with a Multimodal Interface. *Communications of the ACM*, 43(9):45–51, September 2000.

- [80] S. Oviatt. The Human-Computer Interaction Handbook. chapter Multimodal interfaces, pages 286–304. L. Erlbaum Associates Inc., 2003.
- [81] S. Oviatt. The Human-Computer Interaction Handbook. chapter Multimodal Interfaces, pages 286–304. 2003.
- [82] S. Oviatt, P. Cohen, L. Wu, J. Vergo, L. Duncan, B. Suhm, J. Bers, T. Holzman, T. Winograd, J. Landay, J. Larson, and D. Ferro. Designing the User Interface for Multimodal Speech and Pen-based Gesture Applications: State-of-the-art Systems and Future Research Directions. *Human-Computer Interaction*, 15(4):263–322, December 2000.
- [83] S. Oviatt and R. Lunsford. Multimodal Interfaces for Cell Phones and Mobile Technology. *Journal of Sol-Gel Science and Technology*, 8:127–132, 1997.
- [84] A. Paramythis and S. Weibelzahl. A Decomposition Model for the Layered Evaluation of Interactive Adaptive Systems. In *Proceedings of UM2005, 10th International Conference on User Modeling*, pages 438–442, July 2005.
- [85] D. Porta, D. Sonntag, and R. Nesselrath. New Business to Business Interaction: Shake your iPhone and Speak to It. In *Proceedings of MobileHCI 2009, 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, Bonn, Germany, September 2009.
- [86] A. Ramsay, M. McGee-Lennon, G. Wilson, S. Gray, P. Gray, and F. De Turenne. Tilt and Go: Exploring Multimodal Mobile Maps in the Field. *Journal on Multimodal User Interfaces*, 3, 2010.
- [87] L. M. Reeves, J. Lai, J. A. Larson, S. Oviatt, T. S. Balaji, S. Buisine, P. Collings, P. Cohen, B. Kraal, J.-C. Martin, M. McTear, T. Raman, K. M. Stanney, H. Su, and Q. Y. Wang. Guidelines for Multimodal User Interface Design. *Communications of the ACM*, 47(1), January.
- [88] T. Reis, L. Carriço, and C. Duarte. Mobile Interaction: Automatically Adapting Audio Output to Users and Contexts on Communication and Media Control Scenarios. In *Proceedings of the UAHCI 2009, 5th International on Conference Universal Access in Human-Computer Interaction. Part II: Intelligent and Ubiquitous Interaction Environments*, pages 384–393, San Diego, CA, 2009.

- [89] T. Reis, M. de Sá, and L. Carriço. Multimodal Interaction: Real Context Studies on Mobile Digital Artefacts. *Haptic and Audio Interaction Design*, pages 60–69, 2008.
- [90] N. Reithinger, J. Alexandersson, T. Becker, A. Blocher, R. Engel, M. Löckelt, J. Müller, N. Pflieger, P. Poller, M. Streit, and V. Tschernomas. SmartKom: Adaptive and Flexible Multimodal Access to Multiple Applications. In *Proceedings of the ICMI 2003, 5th International Conference on Multimodal Interfaces*, pages 101–108, Vancouver, British Columbia, Canada, 2003.
- [91] J. Rico and S. Brewster. Usable Gestures for Mobile Interfaces: Evaluating Social Acceptability. In *Proceedings of CHI 2010, 28th International Conference on Human Factors in Computing Systems*, pages 887–896, Atlanta, Georgia, USA, April 2010.
- [92] S. Ronkainen, E. Koskinen, Y. Liu, and P. Korhonen. Environment Analysis as a Basis for Designing Multimodal and Multidevice User Interfaces. *Human–Computer Interaction*, 25(2):148–193, 2010.
- [93] L. Rothrock, R. Koubek, F. Fuchs, M. Haas, and G. Salvendy. Review and Reappraisal of Adaptive Interfaces: Toward Biologically Inspired Paradigms. *Theoretical Issues in Ergonomics Science*, 3(1):47–84, 2002.
- [94] G. Salvaneschi, C. Ghezzi, and M. Pradella. Context-Oriented Programming: A Software Engineering Perspective. *Journal of Systems and Software*, 85(8):1801–1817, August 2012.
- [95] C. Sandor. Overview over User Interface Paradigms. University Lecture, 2004.
- [96] G. Schiefer and M. Decker. Taxonomy for Mobile Terminals – A Selective Classification Scheme. In *Proceedings of ICE-B 2008, International Conference on E-Business*, pages 255–258, Porto, Portugal, July 2008.
- [97] B. Schilit, N. Adams, and R. Want. Context-Aware Computing Applications. In *Proceedings of the WMCSA 1994, 1th Workshop on Mobile Computing Systems and Applications*, pages 85–90, Santa Cruz, California, USA, December 1994.
- [98] A. Schmidt, M. Beigl, and H. Gellersen. There is More to Context than Location. *Computers and Graphics*, 23(6):893–901, 1999.

- [99] S. C. Seow, D. Wixon, S. MacKenzie, G. Jacucci, A. Morrison, and A. Wilson. Multitouch and Surface Computing. In *Proceedings CHI EA 2009, 27th International Conference Extended Abstracts on Human Factors in Computing Systems*, pages 4767–4770, Boston, MA, USA, 2009.
- [100] M. Serrano, L. Nigay, R. Demumieux, J. Descos, and P. Losquin. Multimodal Interaction on Mobile Phones: Development and Evaluation Using ACICARE. In *Proceedings of MobileHCI 2006, 8th Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 129–136, Espoo, Finland, 2006.
- [101] R. Sharma, V. I. Pavlovic, and T. S. Huang. Toward Multimodal Human-Computer Interface. *Proceedings of the IEEE*, 86(5):853–869, May 1998.
- [102] B. Shneiderman and C. Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson Addison-Wesley, Upper Saddle River, NJ, 5. edition, 2009.
- [103] S. Singh. Quality of Service Guarantees in Mobile Computing. *Computer Communications*, 19:359–371, April 1996.
- [104] D. Sonntag, R. Engel, G. Herzog, A. Pfalzgraf, N. Pflieger, M. Romanelli, and N. Reithinger. Smart Web Handheld – Multimodal Interaction with Ontological Knowledge Bases and Semantic Web Services. In T. Huang, A. Nijholt, M. Pantic, and A. Pentland, editors, *Artificial Intelligence for Human Computing*, volume 4451 of *Lecture Notes in Computer Science*, pages 272–295. Springer Verlag, 2007.
- [105] S. Tamminen, A. Oulasvirta, K. Toiskallio, and A. Kankainen. Understanding Mobile Contexts. *Personal Ubiquitous Computer*, 8(2):135–143, May 2004.
- [106] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod. Accurate, Low-energy Trajectory Mapping for Mobile Devices. In *Proceedings of the NSDI 2011, 8th USENIX Conference on Networked Systems Design and Implementation*, pages 20–20, Boston, MA, March 2011.
- [107] M. Turk and G. Robertson. Perceptual User Interfaces (introduction). *Communications of the ACM*, 43(3):32–34, 2000.
- [108] M. Turunen, A. Kallinen, I. Sánchez, J. Riekki, J. Hella, T. Olsson, A. Melto, J.-P. Rajaniemi, J. Hakulinen, E. Mäkinen, P. Valkama, T. Miittinen, M. Pyykkönen, T. Saloranta, E. Gilman, and R. Raisamo. Multimodal

- Interaction with Speech and Physical Touch Interface in a Media Center Application. In *Proceedings of ACE 2009, International Conference on Advances in Computer Entertainment Technology*, pages 19–26, Athens, Greece, October 2009.
- [109] A. van Dam. Post-WIMP User Interfaces. *Communications of the ACM*, 40(2):63–67, February 1997.
- [110] L. Van velsen, T. Van der geest, R. Klaassen, and M. Steehouder. User-Centered Evaluation of Adaptive and Adaptable Systems: A Literature Review. *Knowledge Engineering Review*, 23(3):261–281, September 2008.
- [111] R. Vertegaal. Attentive User Interface. *Communications of the ACM*, 46(3):30–33, March 2003.
- [112] R. Wasinger, A. Krüger, and O. Jacobs. Integrating Intra and Extra Gestures into a Mobile and Multimodal Shopping Assistant. In H. Gellersen, R. Want, and A. Schmidt, editors, *Pervasive Computing*, volume 3468 of *Lecture Notes in Computer Science*, pages 323–328. Springer Verlag, 2005.
- [113] M. Weiser. The Computer for the Twenty-First Century. *Scientific American*, 265(3):94–104, 1991.
- [114] P. Wellner, W. Mackay, and R. Gold. Back to the Real World. *Communications of the ACM*, 36(7):24–26, July 1993.
- [115] J. Wesson, A. Singh, and B. van Tonder. Can Adaptive Interfaces Improve the Usability of Mobile Applications. In *Human-Computer Interaction*, volume 332, pages 187–198. Springer Boston, 2010.
- [116] J. Westman and J. Walters. Noise and Stress: a Comprehensive Approach. *Environmental Health Perspectives*, 41:291, 1981.
- [117] J. R. Wiliamson, A. Crossan, and S. Brewster. Multimodal Mobile Interactions: Usability Studies in Real World Settings. In *Proceedings of the ICMI 2011, 13th International Conference on Multimodal Interfaces*, pages 361–368, Alicante, Spain, 2011.
- [118] J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures Without Libraries, Toolkits or Training: A \$1 Recognizer for User Interface Prototypes. In *Proceedings of UIST 2007, 20th Annual ACM Symposium on User Interface Software and Technology*, pages 159–168, Newport, USA, October 2007.

- [119] L. Wu, S. Oviatt, and P. Cohen. From Members to Teams to Committee-A Robust Approach to Gestural and Multimodal Recognition. *Neural Networks, IEEE Transactions on*, 13(4):972–982, 2002.
- [120] A. Zaguia, M. Hina, C. Tadj, and A. Ramdane-Cherif. Interaction Context-aware Modalities and Multimodal Fusion for Accessing Web Services. *Ubiquitous Computing and Communication Journal*, 5(4), December 2010.