FIRMA: A Development Framework for Elderly-Friendly Interactive Multimodal Applications for Assistive Robots

Nikolaos Kazepis[1], Margherita Antona[1], Constantine Stephanidis[1,2]
[1]Foundation for Research and Technology – Hellas (FORTH), Institute of Computer Science
[2]University of Crete, Department of Computer Science
Heraklion, GR–70013, Greece
e-mail: {kazepis,antona,cs}@ics.forth.gr

***Abstract*—The continuous growth of the older population and the progressive ageing of society worldwide bring about the need for new technological solutions for improving independent living, quality of life and active ageing of older citizens. Recent research efforts have focused on incorporating assistive robotic platforms in the elderly's homes under the role of domestic care givers or social companion. Robotic platforms have been around for quite some time, and researchers have been focused on overcoming essential problems that are related to the nature of robotics and their usage in domestic environments. However, since the field of robotics has matured over the last years, a focus shift from the hardware itself to Human Robot Interaction (HRI) in domestic environments is becoming increasingly necessary. This paper focusses on interaction in the context of the collaborative co-existence of the elderly and the robot. In this context, interaction should be tailored to the end users taking into account the specific requirements of each individual, the environmental state but also the capacity of the input/output channels provided by the robotic platform. To this end, this paper proposes a generic platform targeted to support the development of multimodal, elderly friendly, interactive applications that target assistive robots for elderly users.**

Keywords-development framework; multimodal interaction; adaptation; assistive robots.

I. INTRODUCTION

Older people are increasingly becoming the dominant group of customers of a variety of technological products and services (both in terms of number and buying power). Recent advances in Information and Communication technologies (ICT) have great potential for meeting the needs of older people and help them stay healthier, live independently for longer, counteract reduced capabilities due to age, and remain active.

In particular, the field of assistive domestic robotic platforms has been drawing considerable attention in recent years. As opposed to other domestic robotic devices, such as automatic floor cleaners or pure surveillance robots, assistive robotic platforms are designed to provide services to their human users through direct interaction, like displaying information, supporting communication with other people or simply entertaining the users [1].

The primary goal of these robots is to make their older users feel safe and less lonely at home, while enabling and facilitating them in their independent or semi-independent living [2], often in the context of an Ambient Assisted Living environment [9].

Designing and developing appropriate user interfaces for assistive robots presents several challenges due to the demanding target user group and the complexity of the environment.

Multimodal interaction including a graphical user interface, speech input and output, as well as gesture input has been found in various research efforts as an adequate solution for older users to interact with robots [10]. However, at present developing such interfaces is a very demanding task mainly performed ad-hoc, due to the lack of tools and systematic approaches. An additional important need is to support the adaptation of modalities to cater for the target user diversity.

This paper proposes a framework, named FIRMA, to support the development of multimodal, elderly friendly, interactive applications for assistive robots targeted to elderly users in AAL environments. FIRMA provides developers with the necessary technologies, tools and building blocks for creating elderly-friendly multimodal applications in AAL environments, with particular focus on robotic platforms, thus increasing their level of adaptation to users' needs. Using the proposed framework makes these applications inherently friendly to the elder users and capable of adapting to their needs, the surrounding environment and the context of use. The framework facilitates the effective and efficient development of the supported user interfaces, thus simplifying to a great extent the developer's work.

The rest of this paper is organized as follows. Section II discusses the implications of designing for elderly users and overview state of the art research on elderly-friendly multimodal applications for assistive robots. Section III describes the architecture of the FIRMA framework. Section IV discusses FIRMA's implementation. Section V demonstrates a test case application built based on the FIRMA framework. Section VI discusses the evaluation of the FIRMA framework and section VII concludes the article.

II. BACKGROUND

Several user studies have shown that elderly people and their families regard social inclusion, safety and home automation as important features of future homecare environments [11]. With respect to interaction in such

 ISBN: 978-1-61208-468-8

environments, one of the main research challenges is the design of adequate user interfaces. This is due to the fact that elderly people vary considerably in their physical and cognitive abilities, which makes it difficult to use traditional forms of interaction [3]. Focusing on single interaction strategies may not always provide appropriate solutions [4], as many older computer users are affected by multiple functional limitations.

To address this problem various authors developed intelligent user interfaces, which support users according to their individual needs. For example, [25] introduces a spatial metaphor for universal control devices to structure available services based on the elderly person's own apartment. The results of the study showed that the apartment metaphor is actually appropriate to enable elderly people to access a large number of services available in an AAL environment in an intuitive way. The metaphor showed a way for structuring and visualizing services in a universal control device.

Furthermore, [26] presented a novel general framework for multimodal dialogue processing, which is conceived following an application-independent philosophy. In fact, it is able to manage multimodal communication between people and the environment in different application scenarios. The core of the framework architecture is composed of the analysis and planning levels, which enable the processing of information derived from whatever input modalities, giving these inputs an appropriate representation and integrating these individual representations into a joint semantic interpretation.

Moreover, [27] presents a prototype for a Web 2.0–enabled ambient assisted living (AAL) device that offers easy-to-use functionality to help elderly people keep and establish new contacts, find events that match their interests and be aided in sustaining their mobility. The prototype consists of a hardware device for mobile usage which host the desired functionality while being adequate for use by elderly people. An internet tablet was selected accommodating a large touch screen.

If carefully designed, multimodal user interfaces can provide an appropriate solution to cater for the needs of elderly users [12]. The main objective is to achieve interaction as natural as human-human communication, while increasing robustness by means of redundant or complementary information. The selection, activation, deactivation and fusion of the appropriate modalities plays a significant role during human-robot interaction, as it offers the users a fully usable system to interact with as well as adapting to their needs, preferences and to the changing semantic context of the interaction.

Regarding touch based interactions targeted to elder users, several research efforts have provided valuable insights regarding the different aspects of how the respective systems should be designed. The research findings include the optimal inter-key threshold that has to be defined (100ms-150ms) [35], the minimum touch target sizes (8 mm or larger) [36], the significance of employing familiar interactions and behaviors to help in orienting older users with new applications [37], new touchscreen input methods for elderly users with tremor (e.g., swabbing) [38], appropriate touch based gestures [39][40], as well as the benefits of employing multimodal feedback to improve task performance [41].

Furthermore when designing speech recognition systems for the elderly target user group, its heterogeneity should be taken into account. Individual persons have different individual needs based on their different age related health impairments. Such impairments can affect the person's speech capabilities which results either in an increasingly limited vocabulary or fluctuations in their pronunciation clarity [42]. These limitations should be taken under consideration when designing the parameterization properties of such speech engines.

Moreover, the heterogeneity of the elderly user group implies various restrictions when designing and developing gestural interaction modalities for the elder users. Individual persons have different individual needs based on their different age related health impairments. Such impairments can affect the person's mobility capabilities and cognitive functions which results in mobility restrictions in different body parts and difficulties in remembering the specified gestures and the optimal way of performing them. These limitations should be taken under consideration when designing the parameterization properties of such gesture recognition engines [43].

Despite the fact that multimodal user interfaces have been in focus for quite some time [1], and much research has been conducted to address the main challenges of modality interpretation, coordination, parameterization and integration [13], developing multimodal user interfaces is still a difficult endeavor. Various approaches have been investigated to facilitate multimodal user interface development, such as, for example, [14][15][16]. However, these approaches are dependent on the interaction platform and mainly target conventional PCs and mobile devices.

According to [20] recently developed assistive robots, such as ALIAS [17], DOMEO [17], KSERA [18], CompanionAble [19] and HOBBIT [20], despite the differences introduced by the various robotic platforms, have multimodal user interfaces characterized by similar modality options and architectures. The basic offered modalities are touch-based interaction on some screen integrated in the robot, speech input and output, and gestures. A central module, in some cases called "Dialogue Manager", is responsible to control the output based on user input and system state and coordinating the different input and output modalities.

Despite their similarities, all the above mentioned interfaces have been developed ad hoc, as no reference framework currently exist for facilitating developers of multimodal user interfaces for assistive robots. As a consequence, the developed interfaces suffer from lack of flexibility, are difficult to customize, modify and reuse, require cumbersome solutions to communicate with both the ROS operating system [22] running on the robot and the AAL environment, and exhibit limited adaptation capabilities.

Against the above background, the proposed FIRMA framework allows the effective and efficient development of

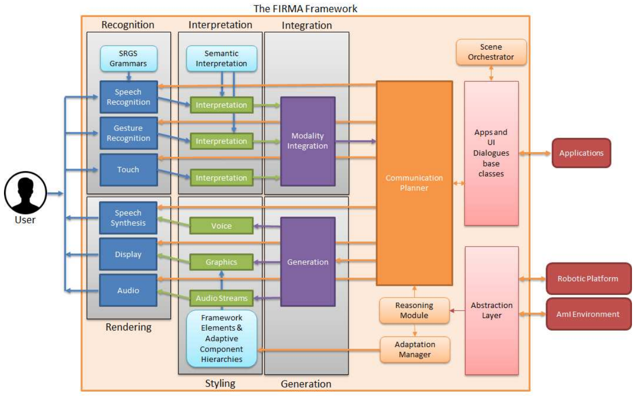

Figure 1. Orchestration of the different conceptual layers

multimodal adaptable user interfaces for assistive robot applications, relieving developers form the burden of programming ad hoc solutions.

III. Architecture

The FIRMA framework comprises a collection of conceptual layers which can be seen in Figure 1.

A. *The interaction recognition layer*

The user is able to interact with the system through the interaction recognition layer. This layer consists of the different available interaction modalities that are provided. Additional interaction modalities can be added in future work such as hardware buttons and switches. The user is able to interact with the robot using touch, gestures and voice. These modalities are adapted to the profile of the user, his preferences and the context of use. They are managed, selected and fused together by the communication planner functional submodule. The touch recognition modality corresponds to the touch interactions between the user and the robotic platform's onboard touch screen.

The gesture recognition modality refers to the set of preselected gestures that the user is able to perform and the robot is able to understand and behave accordingly based on the context of the interaction. Finally, the speech recognition input modality refers to the predefined set of SRGS speech recognition grammars that describe the set of vocal commands that the robotic platform is able to understand. This set of SRGS grammars is loaded into the speech recognition engine so that the robot will be able to interpret the user's speech accordingly.

B. *The input interpretation layer*

The output of the interaction recognition layer is fed into the input interpretation layer. This layer consists of the processing of the user input in term of semantic interpretation based on the context of interaction. Each input modality of the input recognition layer is interpreted accordingly to the profile of the user and the interaction context. The speech recognition modality is interpreted according to the semantic speech annotations that are included in the corresponding SRGS speech grammars. The gesture recognition modality input is interpreted according to the respective application's logic that is active during the interaction, as well as the context of the interaction. For example, the same affirmative gesture may have different interpretations according to the context, and hence it could be interpreted either as a "YES" in the context of a question or as a "NEXT" in the context of an interaction process. Finally, the touch modality input is interpreted based on the dialogue that is displayed at the time that the interaction took place.

C. *The modality integration layer and the low level framework architecture*

The interpreted input is fed from the input interpretation layer into the modality integration layer, where the input from all the different available modalities is integrated based on high level integration scripting. For this purpose, the ACTA runtime (see Section IV.A) is used to integrate all the available modalities into a uniform input channel that can be routed to the communication planner functional component in order to take the necessary decisions regarding the orchestration of the input and output modalities. The same integrated input becomes available to the respective active applications through the low level input mechanisms that the FIRMA framework provides through the base classes that the developed applications inherit. Furthermore, the different available applications can communicate with the scene orchestrator functional component in order to gain access to the functionality it provides regarding the management of the different application screens and their display on the onboard robot screen. The input from the sensors of the robotic platform as well as the input from the environment is transformed into system readable format. The input from these sources is then routed through the reasoning module of the framework in order to infer all the necessary adaptation and communication decisions. When there is need for output from the system to the user, the communication planner decides over the selection and the fusion among the different available modalities to generate the information to be conveyed to the user.

D. *The output styling layer*

The information to be conveyed to the user goes from the generation layer to the output styling layer. This layer is where all the styling over the information delivery takes place. For each one of the available modalities, the appropriate styling is selected according to the user model, his preferences and the context of the interaction. The styling can refer to the output for the speech synthesis modality, the output for the UI display modality or the output for the audio modality. For the speech synthesis modality, the appropriate voice is selected according to the preferences of the user. Additionally, the appropriate rate, volume and pitch of the voice is selected and the output speech is styled using the SSML markup language. For the UI display modality, the appropriate UI selection and adaptation takes place according to the decisions of the adaptation manager functional component. The appropriate UI elements and dialogues are selected, the appropriate component hierarchies are instantiated and the output is delivered to the robot's display for the user to interact with. Furthermore, for the audio output modality, the appropriate auditory feedback is selected and the parameters of the audio output are specified. Finally, the output from the output styling layer is wired to the output rendering layer.

E. The output rendering layer

The final stage of the output delivery is the output rendering layer. This is the layer responsible for delivering the actual output to the users. It comprises the different available output modalities as they have been selected and fused by the communication planner functional component. For the speech synthesis output modality, the actual speech is generated based on the SSML annotations from the styling layer and the final auditory feedback is delivered to the user. For the touch display output modality, the appropriately selected framework elements, components and dialogues are instantiated and the result is presented on the robot's onboard display. Finally, for the audio output modality, the appropriate adaptation parameters are applied and the auditory feedback is delivered to the user.

IV. IMPLEMENTATION

FIRMA is a fully integrated development framework that can support the design and development of elderly friendly, multimodal interactive applications that are deployed on domestic robotic taking full advantage of the possibilities they can offer. The results of this research effort include all the necessary tools and building blocks for the creation of speech enabled, voice recognition enabled, gesture recognition enabled, and touch enabled adaptable and adaptive interactive applications.

The hardware requirements for the FIRMA framework are relatively low. The framework runs under windows 7 or later either 32 or 64 bit and requires a Core 2 Duo or better processor. The touch enabled interactions require a touch screen tablet / laptop or a touch enabled monitor. All the framework components tools and modalities run under Windows on the touch enabled computer except for the gesture recognition modality that runs under Linux on the robotic platform and communicates with the rest of the framework through the ROS middleware.

A. ACTA: A general purpose finite state machine (FSM) description language for ACTivity Analysis

ACTA is a general purpose finite state machine (FSM) description language [5]. ACTA's primary design goal was to facilitate the activity analysis process during smart game design by early intervention professionals who are not familiar with traditional programming languages. However, developers can use ACTA also for applications whose behavior is composed of a finite number of states, transitions between those states and actions, as well as for application based on rules driven workflows. The ACTA runtime mechanism provided the base on which the framework's reasoning and adaptation mechanisms were built. ACTA's Runtime has been adopted and adapted to fit the needs of the creation of Multimodal interactive Applications (ARMA).

B. ARMA: Extending ACTA Runtime to support the development of Multimodal elderly friendly Applications

ACTA's runtime is based on the Windows Workflow Foundation framework (WWF). The ACTA IDE is used to code all the application interaction logic which can then be extracted to an XML rules file for further use. The rules file can be loaded into a WWF Rule Engine which is an event driven reasoning engine that can run the provided rules and conclude to the desired actions and transitions between the different states of the application's logic.

The main workflow for creating an interactive application includes the definition and design of its different screens and then the definition of its various states. Usually, one state is then mapped to one application dialogue screen. However, states with no visual output can exist, and UI dialogue screens can map to more than one different states of the application.

C. Loading and unloading rules at runtime

A very useful functionality that has been added to the ACTA backend in ARMA is the option to load and unload rules at runtime. This contributes to the reduction of the rules that are loaded at any given time. Furthermore, it offers the ability to change the behavior of the developed applications based on the subset of rules that are loaded at a given point in time. This enables the use of abstract task hierarchies that can be instantiated at runtime, while the respective rules that support their functionality are loaded at runtime. Furthermore, this addition opens new paths for adaptation based on the extra subset of loaded rules. For example, the experience of the user can be taken into account when he/she is expected to fulfill specific tasks and the UI that he/she is presented with can change accordingly. Moreover, tasks that are frequently required are automatically adapted and embedded into the framework.

The dynamic loading and unloading of rules has been implemented in full compliance with the functionalities of the language for rule activation/deactivation, rule prioritization etc. The backend has been extended to support the dynamic rule loading by respecting the aforementioned properties and treating them appropriately. Since the WWF does not provide the necessary functionality for merging rulesets, the whole process of the dynamic rule loading was added. Loading and unloading extra rules as needed is more convenient than having all the rules loaded at all times and then activating or deactivating a subset of them as desired, since the latter approach can have a huge performance impact on the whole rule engine (which would have to linearly browse through the whole ruleset to find the respective active rules) and was thus avoided. During the loading of new rules, the rule engine is temporarily paused and the new rules are appended to the currently active ruleset. The old pre-existing rules are not removed or disabled because their functionality is still needed as the new rules do not substitute the old ones but merely temporarily extend the functionality of the application. After loading the new rules, the back-end ACTA data structures are augmented accordingly to support the rule addition without affecting the language's mechanisms such as the mechanism for dynamic rule activation and deactivation or the capability for rule prioritization. Upon a successful append, the rule engine is resumed to activate the functionality that is offered by the new rules. Finally, when the functionality that is offered by the new rules is no longer needed, they are unloaded and the back-end data structure changes are reverted.

When a new set of rules is loaded, it is validated against the rule engine and then run against the instance of the application. The validation is always successful because all the function calling and property manipulation of the rules is implemented through a set of auxiliary helping functions. This functionality is inherently embedded into the ACTA language so that the produced ruleset is transcribed using these functions. This approach has the advantage that most of the fatal conditions can be silently ignored with the corresponding error messages being printed on an error log file while the state of the application remains stable. This means that if the ACTA script contains instructions for calling functions or setting properties that can't be found neither in the framework base classes nor in the developer-created derived instances, the invocation of those functions can fail silently without compromising the stability of the whole system.

D. Modality integration

Modality integration has been realized by leveraging the different modality generated events and consolidating them at a higher level where the corresponding application can treat them appropriately. This was achieved by implementing various mechanisms in the ACTA backend and in the frameworks base classes.

The framework contains backing fields for modality events. The ACTA backend was extended accordingly to support these fields. When the user interacts with the UI using touch events and touch gestures, these interactions are interpreted into the corresponding events and transferred to a higher level inside the application. For example, when the user presses a button, it generates an event in the base class of the application which is part of the framework. The base class contains the rule engine that can run the loaded ruleset against such events. The user is then able to interact with the UI based on the functionality that has been coded into the application's ACTA script. The result of the activation of the different rules includes state changes and UI dialogues activation in the derived application classes. This way the sequence of the application's dialogues can be easily tweaked and rearranged by the developer as needed.

A very useful feature of the FIRMA framework is the functionality it provides for modality integration at two different levels. The various available modalities can be integrated in the scope of an application's dialogue screen where the developer has to cater for each of the available modalities' events and act accordingly. Another approach would be the consolidation of the modalities into a single one and then develop a corresponding modality handling script that caters for this consolidated modality. Furthermore, modality consolidation can happen either in the scope of an application's dialogues or in the higher scope of the ACTA logic. For example, if the user can issue a command by touch, voice or gesture, the different modalities could be consolidated into the button press in the scope of the application dialogue or in the scope of the ACTA scripting logic which is at a higher level. The developer then could only cater for the single touch press modality as the other two modalities would automatically get consolidated into the touch modality scope.

Taking the modality events and raising to a higher level where they can be easily handled by the ACTA script contributes to the modular nature of the proposed framework's architecture, as the framework's components are loosely coupled and completely asynchronous. Additional modalities such as hardware switches and different kinds of sensors and actuators can be incorporated into the framework with minimal effort, extending the provided functionalities and conforming to the user's needs.

E. Interaction Modalities

The modalities that have been developed and integrated into the proposed framework range from speech recognition and synthesis to gesture recognition and touch interaction. They all have been developed to be fully extensible and configurable both at startup and at runtime so that they can change to reflect the changing needs of the users or the dynamically changing factors of the surrounding environment e.g., ambient lighting, environment noise, active electric appliances etc. In addition, the configurable parts of the developed integrated modalities have been offered as ROS services to the system to support dynamic adaptation based on interaction logic that runs on the robotic platform.

1) Speech Recognition Modality

Speech is an effective and natural way for people to interact with applications, complementing or even replacing the use of mice, keyboards, controllers, and gestures. A hands-free, yet accurate way to communicate with applications, speech lets people be productive and stay informed in a variety of situations where other interfaces would be difficult to use.

The implemented speech recognition modality engine supports adaptation based on the distance between the robot and the user, the vocabulary and the variety of the individual equivalent commands that can be used by the users and understood by the system, the semantic interpretation of recognized commands and the recognition confidence threshold. Furthermore, it supports the dynamic activation of both plain text and compiled speech recognition (SRGS) grammars and is accessible through a ROS node to the rest of the system.

2) Speech Synthesis Modality

The speech synthesis modality of the FIRMA framework has been based on the speech engine functionality provided by the Microsoft Speech Synthesis namespace. This namespace contains classes that offer the initialization and configuration of a speech synthesis engine, the creation of prompts, the generation of speech, and the modification of the synthesized voice characteristics. Speech synthesis is often referred to as text-to-speech or TTS.

The implemented speech synthesis modality engine can be tailored to the needs and preferences of the users as well as the context of the interaction by offering adaptation parameters exposing the gender of the used voice, the speech volume, the rate as well as the pitch of the generated output.

3) Gesture Recognition Modality

Gesture recognition is the process by which gestures made by the user are made known to the intelligence system. Gesture recognition plays a significant role in Human Robot Interaction since it adds a natural dimension to the interaction process. People inherently use their hands when talking to convey their thoughts, intentions and feelings. Providing robotic platforms with a way to understand this kind of body language, opens new dimensions for intelligent household robotics that can understand their user more accurately.

The gesture recognition modality has been integrated into the proposed framework. The recognition engine that has been developed to cover the gesture modality needs of the proposed framework is able to understand a predefined set of gestures that are relatively easy to perform and be remembered by the end users of the platform. FORTH's gesture recognition module [6][7] has been used to this end. This gesture recognition module is subdivided into three submodules, a submodule capable of tracking the upper body joints, a submodule capable of tracking the person's full body, and a submodule for tracking the hands and fingers of the person.

4) Touch Modality

The touch modality refers to the interaction that takes place between the human and the touchscreen tablet pc that is onboard household robotic platforms. Touch is an important aspect of human robot interaction because it consists a natural human approach. Selecting between desired items, reaching for different types of controls and adjusting various sensors are all part of humans' daily lives. The simulation of such daily activities can be done by using a touchscreen tablet PC that can be used both for output and input form the users to the robotic platform.

The proposed framework integrates all the aforementioned modalities into a seamless set of interaction modes between the robot and its users. This results into a more natural form of interaction, since the user is free to choose how to interact with the system based both on his/her preferences and the context of interaction. The robot can display its output on the onboard touchscreen device and use sound at the same time as redundant auditory feedback just like when people interact with each other. Furthermore, the robot is able to understand touches on the touchscreen device, gestures in front of the monitoring image acquisition sensors as well as speech commands given by the users. This provides redundant feedback which has been proved to be necessary especially when designing for the elderly user group [8].

Regarding the Graphical User Interfaces that are being produced based on the proposed framework, they are tailored to the needs of the end users.

The framework's building blocks have been designed based on the user-centric design principles and based on simplicity and clarity of the individual modes that each module represents (e.g., time selection module, binary decision module, multiple selection option module, etc).

Furthermore, the used vocabulary can be easily adapted to the cognitive abilities of the users. The generated UIs are inherently translated into the user's native languages in the sense that the translation files are automatically generated by the framework and the developers are only required to provide the literal translation of the set of sentences that they are being given into the end-users' native language. In other words, the produced user interfaces are globalization and localization ready since the necessary language translation files are automatically generated by the system at runtime and can be edited offline.

Finally, the framework provides quick exit shortcuts to the main menu and access to emergency scenarios.

F. Adaptation

The different modalities that are supported by the framework can be activated or deactivated individually according to the preferences of the users and the context of interaction. The framework can decide on the optimal set of modalities to enable, fine-tune and fuse together in order to provide the end users with an interaction as seamless and as natural as possible. Furthermore, the selection of the different modalities and their fusing is transparent to the developer, as it is handled automatically by the framework.

The developer has full control over which modalities are going to be supported at any given time as well as when and how they will be activated or deactivated. However, the developer is also given the opportunity to provide the basic functionality that he wants to make available to each of the aforementioned modalities and then let the framework decide on how and when each modality gets activated. For example, the developer can explicitly specify which parts of his application can benefit from a specific modality and which parts must be contained only to specific modes of interaction. He can specify when he wants only a specific modality to be used or when any input from any of the available modalities can be considered valid. For example, he can enforce that for critical application decisions, only the touch modality will be considered a valid way of confirmation, while for all other parts, any speech or gestural input will be allowed to be interpreted and treated accordingly. Finally, the framework is able to handle tricky cases where one modality might have to be deactivated due to dynamically changing conditions, although the developer has allowed its input. For example, the speech modality might have to be deactivated in noisy environments, or the gesture recognition modality might have to be deactivated in situations where the environment light is insufficient.

FIRMA supports adaptation through both adaptive component hierarchies and adaptive style hierarchies. The former is based on the design and implementation principles of unified user interfaces [21], while the latter is based on the use of adaptive style hierarchies as they are supported by the Windows presentation framework, to either specify the desired application coloring scheme and sizing guide for the different controls and UI elements, or change completely the different framework elements' appearance.

Adaptive component hierarchies are inherently supported by the proposed framework. Tasks are described in an abstract manner at a higher level using ACTA, while general guidelines are provided according to their instantiation strategies. For example, the time selection task can be declared to comprise the consequent selection of hours, minutes and time specifiers according to the time of the day. General guidelines can be stated according to the expertise of the user encoded in his profile. These guidelines specify how the whole task of time selection can be orchestrated in order to be presented to the user who is going to be guided through the process of time selection. Furthermore, user preferences are taken into account, so that specific user control are used or omitted during the process. Finally, the entire task is realized in a transparent to the developer manner who can simply declare that he needs the time selection process at the desired place inside the applications that he builds. The initiation of the task takes place automatically, and the developer can explicitly declare the starting and ending state and consequently the starting and ending application dialogue that will be displayed to the end user.

In addition to the adaptive component hierarchies' principles and design guidelines, the approach of adaptive style hierarchies has been adopted. According to this approach, the sizes and colors of the displayed framework elements can be controlled by styles that can be applied both at design time and at runtime. A number of cascading stylesheets have been developed to be used in the context of adaptation based on this approach. A subset of the developed styles have been used during design time so that the developer can have a clear understanding of the appearance of the different user controls and dialogues that he/she is incorporating into the developed applications. The design time styles collection has been consolidated into a single higher level style file which can be included in the designed user controls and dialogues.

The adaptive style hierarchies that are used for adaptation purposes during runtime have been split into three major categories. The first contains all the styles that handle how the different framework elements will be displayed. These styles contain all the individual stylistic decisions that drive the appearance and define the visual tree of all the framework elements such as buttons, lists, dialogues, text entry controls, labels etc. The second category contains all the styles that define the coloring scheme of the application including foreground and background colors for all framework elements, border brushes of the different user controls, darker backgrounds for giving emphasis to specific UI elements, etc. Finally, the third major category contains all the styles that correspond to the sizing decisions of all the framework elements and UI dialogues, including button sizes, dialogue sizes, virtual keyboard sizes and margins, text input control sizes, etc. The appearance of the final user interface is decided at runtime by the adaptation manager through a process of "pick and match" among the different available cascading adaptive style hierarchies, by selecting one from each major category. As a result, one style for visual appearance is selected, one style that defines the coloring scheme is placed on top of that and finally one more style that defines the overall sizes of every element is superimposed on the selection for filling in the missing sizing information and restoring the dynamic bindings between all three style collections. As a result, every style can refer to any other category of styles through the use of dynamic resources declarations. This means that each style is only responsible for its own category while being allowed to contain bindings across different categories. Hence, the visual appearance styles can contain bindings to the sizing category styles which are going to be realized once the specific sizing resource dictionary that is going to be used, has been defined and linked to the runtime of the framework. This approach can create an arbitrary number of application appearances based on the selection of the activated styles and the possible combinations among them. For example, if there are three different styles defined in each of the three different major style categories, the developers can choose among any of the twenty seven combinations (i.e., 3x3x3) of the available UI instantiations. However, the selection of the developers are being superseded by the adaptation manager decisions as deemed necessary at runtime.

G. Globalization and Localization

The proposed framework provides inherent support for globalizing and localizing the developed applications to the native language of the users. The supported globalization functionality is provided by supporting the automatic generation of the necessary translation files. The developers are only required to edit these files to provide the literal translations of the provided phrases in the end-users' native language. The automatic translation is supported for all the framework user controls, dialogues and elements that are being used. The localization of the developed applications is realized by means of a universal translator auxiliary helper class that has been developed as part of the framework. The localization is based on localized culture and locale specific resource files that can be translated by either the developer or by expert translators to the end user's native language.

One major point of the translation module is that all translations are based on keys which can be prefixed with any desired phrase. The translation mechanism was designed having in mind that each translated string should have a corresponding key which could be prefixed by the fully qualified name of the assembly that the translated element belongs to, followed by the name of the application that contains the element. However, when a translated control belongs to a specific application dialogue, the name of the respective dialogue is used instead of the application name.

V. THE ALARM CLOCK APPLICATION TEST CASE

To demonstrate the functionality and the effectiveness of the FIRMA framework, this section presents a sample multimodal application developed using FIRMA. It is an alarm clock application that can be used for managing a user's daily tasks scheduled for specific times of the day. The user can use the application to see the current time, see daily notifications, add new alarms, delete existing alarms and snooze elapsed alarms. The application supports speech recognition and synthesis, gesture recognition, touch enabled interactions and is adaptable and adaptive to fit the needs of the users.

The adaptations that have been implemented for this test case application concern the coloring scheme of the application, which changes depending on the level of the ambient lighting in the surrounding environment, and the size of the used controls, dialogues and messages with respect to the relative position of the user and the distance between the robot and the user. When the lighting level of the room increases, the coloring scheme of the application changes to darker colors that have higher contrast for the user to be able to see more clearly. Furthermore, when the level of ambient lighting is reduced, the application automatically changes into a more vibrant color scheme to compensate for the lighting changes. When the user is seating, the size of the used controls, dialogues and messages adapt according to the distance between the user and the robotic platform. The application supports three different sizes, a large sized scheme for bigger distances, a medium sized scheme for average distances and a small sized scheme for a more comfortable interaction when the robotic platform is very close to the user. Furthermore, the application supports a dark colored scheme for the night and a light colored scheme for the day. Moreover, when the robot detects that the user is not wearing his/her glasses, the application's scheme changes to a high contrast coloring scheme for convenience. Figure 2 shows the different coloring and sizing schemes that the alarm clock application supports.

According to Figure 2, on the lower bottom right corner of the dialogues, a green visual cue representing the status of the speech recognition modality can be seen.

The speech recognition modality is active, hence the green "ear" icon is visible. The Home button of the main UI Navigator window gets enabled whenever the user navigates away from the home screen of the main menu. Whenever the robot speaks, the speech recognition modality gets deactivated to prevent the robot from understanding its own speech as commands to itself. The deactivation of the speech recognition modality is represented by a red "ear" image with an accompanying strike-through diagonal line.

Furthermore, the speech synthesis modality is represented by a similar visual cue which depicts an orange robotic face figure which animates when the robot talks.

The bottom right dialogue that is shown in Figure 2 shows the alarms screen of the alarm clock application. In the middle, the user can see a list containing all the daily alarms that are active for the respective day. For each alarm, the time of the alarm and an assigned message that describes it is being displayed. Existing alarms can be deleted and new alarm can be added.

Figure 2. The different coloring and sizing schemes supported by the alarm clock application

Similarly to the alarms screen, the alarm adding screen of the application displays the current time of the system to facilitate the user when he wants to add an alarm at a relatively short time span. The time selection process is automatically tailored to the end user while the respective adaptive task hierarchy is instantiated step by step. The time selection process is adapted to the experience of the user. Different controls and additional steps can be automatically selected by the framework for average or inexperienced users. The user can select the desired time and then accept the changes or reject them to return to the previous dialogue. The user is able to cancel at any time, or press the home button to return to the main menu screen.

The user is given functionality to add a message to be assigned to the alarm. A virtual on-screen keyboard is provided to input the messages text. Finally, the application includes UI dialogues for confirming and providing feedback on the deletion of an alarm and for acknowledging or snoozing elapsed alarms.

VI. FRAMEWORK EVALUATION

The FIRMA framework was evaluated both in terms of being easily and effectively usable by the developers and in terms of being capable of building elderly-friendly applications by means of a heuristic evaluation. This section describes these two different kinds of evaluation. Further evaluation regarding the elderly-friendly aspect of the framework is yet to be conducted in the context of the European RAMCIP project trials as discussed below.

A. Developer based Evaluation

The FIRMA framework was evaluated by developers regarding its efficiency and its ease of use while building elderly-friendly multimodal interactive applications.

 ISBN: 978-1-61208-468-8

Given the target user group of the tool, i.e., developers, who are by definition expert users, it was decided to combine user satisfaction measurement with expert user interface evaluation in order to obtain detailed comments and suggestions on the FIRMA development framework as well as its interface design regarding the ready-made components and framework elements.

The IBM Usability Satisfaction Questionnaires [24] was adopted for subjective usability measurement. The FIRMA framework was evaluated by six expert users with substantial experience in application development. All users had at least a University degree in Computer Science or related subject. All of them had at least a few years' experience in the field of creating WPF applications using the C# programming language and some basic knowledge, but no extensive experience or practice concerning adaptation or localization practices or multimodality approaches. The user group consisted of four males and two females, whose age ranged from twenty-five to thirty-five years.

The group of users was briefly introduced to the main objectives of the FIRMA framework and of the evaluation experiments, and was provided with a brief introduction to the setup of the development environment, a brief description of the FIRMA framework's functionality and tools, and a brief scenario (including an accompanying tutorial) involving the creation of a new toy application that consisted of two dialogue screens, as well as the integration of different modalities, adaptive tasks, adaptation and localization, in order for the developers to be able to perform a more extensive testing of the system's features.

The developers were then requested to perform the tasks in the scenario and fill-in the user satisfaction questionnaires, as well as an expert evaluation report as detailed as possible. The scenario A included the creation of a new basic application while the Scenario B included the integration of the multiple modalities, the localization of the application and the introduction of a few navigation restrictions through the Communication Planner submodule.

The results of the user satisfaction measurement are reported in Table 1 (ASQ) and Table 2 (CSQU). Scenario A showed a variance of 0,019 which resulted into a standard deviation of σ=0.1384, while scenario B showed a variance of 0,056 and a standard deviation of σ=0.2380.

The conduct of the "Create a basic new application" scenario appears from the results to have been easier than the conduct of the "Integrate multimodality, localize it and add restrictions" scenario. This is probably due to the need of developers to acquire some experience in how the framework works, what functionality it offers and how this functionality can be achieved.

TABLE I. AFTER-SCENARIO QUESTIONNAIRE (ASQ) RESULTS (RANGE FROM 1 - HIGHEST - TO 7 – LOWEST)

	User 1	User 2	User 3	User 4	User 5	User 6	Average
Scenario A	2,3	2,5	2,1	2,2	2,3	2,1	2,25
Scenario B	3,7	3,2	3,6	4,0	3,6	3,5	3,60

The most appreciated aspects of the system were found to be its ease of use and overall effectiveness in the context of multimodality integration, automatic adaptation and localization and the reflection of the appearance of the end result during the design and development time in the context of creating elderly-friendly interactive multimodal applications. The users found the required workflow for the creation of new apps to be pleasant and intuitive and they were pleasantly surprised by the different supported automations that were supported by the system "out of the box" such as the multimodality integration, adaptation and localization processes. Concerning the included user interfaces and dialogues of the FIRMA framework, the users found that they are self-explaining, and that the dialogue screens do not contain information that is irrelevant. The users also appreciated the fact that the framework is carefully designed to prevent common problems from occurring in the first place (such as the automatic inclusion of the design time style sheets which reflect the appearance of the end product), and makes dialogues, actions, and dependencies visible. The developers particularly liked the decoupling between the application dialogues and the application logic and were enthusiastic about the fact that fine tuning of the application logic can be done at a higher level without requiring the recompilation of the entire application code. Error messages were also considered to be clear and precisely indicating the problem at hand. Furthermore the users offered helpful comments towards enhancements which are discussed later in this section.

The identified weak points of the framework mainly concerned the limited documentation provided. This was a known shortcoming of the prototype system, attributed to existing constraints at development time, leading to rather limited and focused documentation. The provided tutorial and documentation was focused on the parts of the workflow at which the developers were expected to have the least experience.

As already mentioned, the developers were also requested to provide an expert evaluation report accompanying the filled-in questionnaires. In these reports, the users offered their overall comments as well as more detailed suggestions for improvement of the FIRMA framework. The overall attitude of the users towards the system was positive. It was also pointed out that the tool presents a low cognitive load, and employs workflows and concepts familiar to application developers. However, it was also observed that that developers had to maintain and meddle with different

TABLE II. COMPUTER SYSTEM USABILITY QUESTIONNAIRE (CSUQ) RESULTS

	User 1	User 2	User 3	User 4	User 5	User 6	Average
SYSUSE	2,2	2,3	2,3	3,0	2,1	2,1	2,33
INFOQUAL	3,3	3,2	3,3	3,1	3,6	3,6	3,35
INTERQUAL	3,0	3,1	2,0	2,8	2,0	1,2	2,35
OVERALL	3,1	2,2	2,5	3,0	2,5	3,0	2,72

technologies to make an application work, something that was an expected forthcoming stemming from the nature of multimodal applications.

The developers pointed out that there are some parts of the workflow that could be made further error-proof by providing some additional tools and editors. For example, the developers found the ACTA scripting language rather enjoyable but almost all of them commented that they would like some kind of auto-completion and some code snippets that could expand to provide some skeleton code for creating an additional application state or a transition between the current state and the rest of the states of the current application. Furthermore, they pointed out that the translation of the ACTA scripts into WWF rulesets should be something that should be addressed by the framework itself automatically to avoid synchronization error between the rulesets and the source script files. This was a unanimous request. Moreover, the developers suggested that other parts of the workflow such as the creation of SRGS grammars or the creation of restriction rules in the Communication Planner required adding code in XML which was not very convenient for all of them in respect to their experience with the language. In particular, the majority of them suggested that an SRGS editor should be provided to minimize user errors during the creation or the localization of SRGS grammars. Furthermore the development of an additional editor was advised towards supporting the creation of restriction rules for the Communication Planner while taking advantage of the semantics of the ACTA language which could provide automatic listing of all the available states, dialogue names and transition triggers. In addition, the developers suggested the creation of automation projects for the required project types, class types and dialogue types in the Visual Studio IDE which was a foreseen request since the developers of the actual prototype system had already contemplated on the provision of such functionality in a subsequent version. Other comments concerned limitations and bugs of the current implementation (e.g., window resizing problems, lack of some confirmation dialogues, etc.).

In general, the developers stressed that the availability of such a framework would in their opinion be very helpful in creating elderly-friendly multimodal interactive applications easily and effectively. However, it was also noted that a certain degree of familiarity with the framework needs to be acquired before effective use in real development cases, particularly in relation to the order of the tasks that the user has to perform, which may not be clear at a first glance. Furthermore, some of the users had specific requests for additional functionality and system capabilities they would like to see supported in future versions of the framework. These mainly concerned the inclusion of additional modalities, the formalization of the communication protocol between the framework and the ROS operating system, the provision of automation class types in the Visual Studio IDE for creating ROS nodes and subscribers and expandable code snippets for adding application dialogues in code behind.

In general, the user evaluation of the FIRMA framework offered valuable insights into the functional and the interaction characteristics of the system and reinforced the belief that there is an actual need and demand for a framework providing the building-blocks and tools to support the design and development of elderly-friendly interactive multimodal applications for assistive robots.

B. Heuristic Evaluation

In order to get an initial insight into the elderly-friendliness of the applications created using FIRMA, a heuristic evaluation was performed in order to evaluate the various developed user controls and supported appearance styles. The results of this heuristic evaluation will be enriched with the planned user evaluation that will take place in the context of the EU funded RAMCIP project during spring 2017.

1) Methodology followed

In order to evaluate the UI controls of the FIRMA framework the following procedure was followed:

- A UI window was developed in order to host a demonstrator application for the evaluators
- From the UI window a list of buttons became available to the evaluators so as to select the UI control to evaluate
- By pressing one of the buttons a new window was opened displaying on the center the control to be evaluated
- A drop down menu was available to the evaluators that contained a number of pre-defined profiles. The selection of an option from the drop down menu resulted to the adaptation of the user control to the selected profile

This process was preferred mainly because it made easier for the evaluators to mark the identified usability errors by just filling in a table the name of the control, the selected profile and the error.

For performing the evaluation three usability experts used the presented application and through the application inspected all the available controls and recorded the identified usability problems. These problems were gathered per control and graded based on their severity.

2) Discussion

The results of the evaluation were rather positive in terms of the overall acceptance of the framework by the evaluators and all the identified usability errors were clearly defined and documented. Based on the feedback received, the controls were redesigned and fine-tuned. This preliminary evaluation should be considered as an intermediate step in the overall process of evaluating the outcomes of this research work.

The final evaluation will take place in the context of the European RAMCIP project trials that will take place during a six month period in Spain (Barcelona, Fundacio ACE, Barcelona Alzheimer treatment and research center) and in Poland (Lublin Medical University). The end users will be healthy elderly volunteers and patients with mild cognitive impairments or early Alzheimer Disease. They will interact with the RAMCIP robotic platform in controlled environments during the time period April 2017 – October 2017. The UI of the RAMCIP platform will be based on the FIRMA framework and will be evaluated and validated in terms of ease of learning and ease of use, comfortable perception, acceptability and satisfaction.

All the applications that will be deployed on the RAMCIP robot will be based on the FIRMA framework. For example, a phone dialing application will be developed to enable the elderly user to place phone calls to their friends and relatives. This application will employ image buttons which will be mapped to pre-installed contact details so that the user will be able to place the desired phone call simply be pressing the respective image of the relative that he/she wants to call. FIRMA already provides multiple modality activated picture buttons that can be used by the developers for this purpose, leveraging the burden to integrate the activation of the buttons using all the different available modalities.

VII. Conclusion and Future Work

This paper has presented the FIRMA framework for the development of multimodal adaptable user interfaces for assistive robots. The framework supports modality selection, adaptation and integration, intercommunication with the ROS operating system and the AAL environment, and offers globalization and localization facilities.

The conducted evaluation has shown that the framework can significantly help developers in easily and efficiently creating elderly-friendly multimodal interactive applications for assistive robots. The preliminary heuristic evaluation of the framework has also suggested that the developed applications are inherently elderly-friendly because of the design of the FIRMA's ready-made controls and UI elements.

FIRMA constitutes the primary platform for the development of the user interfaces for the assistive robot under development in the context of the RAMCIP project funded by the European Commission under the HORIZON 2020 Programme. The robot is targeted to support elderly people with mild cognitive impairments. The validity and the effectiveness of the framework will be tested both in the lab and in real life scenarios in the pilot trials of the project which will take place simultaneously in two different European cities over a time span of about 6 months.

Two main directions of further work are anticipated in a path towards supporting the fruition of domestic assistive platforms for the elderly in AAL Environments. The first is the further enrichment and development of the system into a mature product and the second is the adaptation of the framework to cover the needs of different user categories and impairments as well as their families' and caregivers'.

Towards this end, a number of improvements are planned focusing on the currently available modalities as well as the overall system functioning in terms of performance, offered functionality and ease of use for software developers.

Ackwnoledgments

Part of this work has been conducted in the context of the Project ICT-RAMCIP "Robotic Assistant for MCI Patients at home", funded by the European Commission under the Horizon 2020 Framework Programme (Grant Agreement 643433).

References

[1] C. Schroeter et al., "Realization and User Evaluation of a Companion Robot for People with Mild Cognitive Impairments," in Proc. IEEE Int. Conf. on Robotics and Automation (ICRA 2013), Karlsruhe, Germany, 2013, pp. 1145-1151, IEEE 2013, Karlsruhe, Germany.

[2] M. Vincze, "HOBBIT - Towards a robot for Aging Well," Workshop on Human Robot Interaction (HRI) for Assistance Robots and Industrial Robots, May 6-10, ICRA 2013, Karlsruhe, 2013.

[3] A. Jacobs and J. Pierson, "Walking the Interface: Domestication Reconsidered," Brave New Interfaces, 2007, pp. 205-215.

[4] P. De Hert and E. Mantovani, "Intelligent User Interface". Working Document of the Project "SENIOR: Social Ethical and Privacy Needs in ICT for Older Citizens: A Dialogue Roadmap," 2008.

[5] E. Zidianakis, "Supporting Young Children in Ambient Intelligence Environments," Heraklion, 2015.

[6] D. Michel, K. Papoutsakis and A. Argyros, "Gesture recognition for the perceptual support of assistive robots," in International Symposium on Visual Computing (ISVC 2014), 2014, pp. 793-804, Las Vegas, Nevada, USA

[7] D. Kosmopoulos, K. Papoutsakis and A. Argyros, "Segmentation and classification of actions in the context of unmodeled actions," in British Machine Vision Conference (BMVC 2014), Nottingham, UK, 2014.

[8] C. Leonardi, A. Albertini, F. Pianesi and M. Zancanaro, "An exploratory study of a touch-based gestural interface for elderly," in Proceedings of the 6th Nordic Conference on Human-Computer Interaction Extending Boundaries - NordiCHI '10, 2010, pp. 845-850

[9] S. Payr, F. Werner and K. Werner, "AAL Robotics: State of the Field and Challenges," in EHealth2015–Health Informatics Meets EHealth: Innovative Health Perspectives: Personalized Health, vol. 212, 2015, p. 117.

[10] P. Mayer, C. Beck, and P. Panek, ""Examples of multimodal user interfaces for socially assistive robots in Ambient Assisted Living environments," in IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom), 2012, pp. 401-406.

[11] C. Röcker, M. Ziefle, and A. Holzinger, "Social Inclusion in Ambient Assisted Living Environments: Home Automation and Convenience Services for Elderly User," in Proceedings of the 2011 International Conference on Artificial Intelligence, ICAI 2011, 2012, vol. 1, pp. 55-59,.

[12] A. B. Naumann, I. Wechsung, and J. Hurtienne, "Multimodal interaction: A suitable strategy for including older users," in Interacting with Computers, 22(6), 2010, pp. 465-474.

[13] B. Dumas, D. Lalanne and S. Oviatt, "Multimodal interfaces: A survey of principles, models and frameworks," in Human Machine Interaction, 2009, pp. 3-26, Springer Berlin Heidelberg.

[14] J. Bouchet, L. Nigay, and T. Ganille, “ICARE software components for rapidly developing multimodal interfaces,” in Proceedings of the 6th international conference on Multimodal interfaces, pp. 251-258. ACM.

[15] M. Serrano, L. Nigay, J. Y. L. Lawson, A. Ramsay, R. Murray-Smith, and S. Denef, “The openinterface framework: a tool for multimodal interaction,” in CHI'08 Extended abstracts on human factors in computing systems, pp. 3501-3506, ACM.

[16] F. Flippo, A. Krebs and I. Marsic, “A framework for rapid development of multimodal interfaces,” in Proceedings of the 5th international conference on Multimodal interfaces(ICMI '03). ACM, New York, NY, USA, pp. 109-116. DOI=http://dx.doi.org/10.1145/958432.958455

[17] S. Goetze et al., "Multimodal Human-Machine Interaction for Service Robots in Home-Care Environments," in Proceedings of the 1st Workshop on Speech and Multimodal Interaction in Assistive Environments, Association for Computational Linguistics, 2012, pp. 1-7

[18] R. H. Cuijpers et al., “Human robot interactions in care applications”, Gerontechnology 2012; 11(2), pp 353-354 doi: http://dx.doi.org/10.4017/gt.2012.11.02.186.00.

[19] A. Badii et al. “CompanionAble: An integrated cognitiveassistive smart home and companion robot for proactive lifestyle support”, Gerontechnology 2012; 11(2), pp. 358.

[20] P. Mayer and P. Panek, “Towards a multi-modal user interface for an affordable Assistive Robot. In Universal Access in Human-Computer Interaction. Aging and Assistive Environments, 2014, pp. 680-691, Springer International Publishing.

[21] A. Savidis and C. Stephanidis, “Unified user interface design: designing universally accessible interactions,” in Interacting with computers, 16(2), 2004, pp. 243-270.

[22] M. Quigley, et al., “ROS: an open-source Robot Operating System,” in ICRA workshop on open source software (Vol. 3, No. 3.2, p. 5).

[23] M. Jackson, S. Crouch and R. Baxter, “Software Evaluation: Tutorial-based Assessment,” software Sustainability Institute, Edimburgh, United Kingdom, 2011.

[24] R. J. Lewis, “IBM Computer Usability Satisfaction Questionaires: Psychometric Evaluation and Instructions for Use,” in International Journal of Human-Computer Interaction, vol. 7, no. 1, 1995, pp. 57-78.

[25] S. Adam, K. Ssamula Mukasa, K. Breiner and M. Trapp, “An apartment-based metaphor for intuitive interaction with ambient assisted living applications,” in BCS-HCI '08: Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 1, 2008, pp. 67-75.

[26] A. D'Andrea, A. D'Ulizia, F. Ferri and P. Grifoni, “A multimodal pervasive framework for ambient assisted living,” in PETRA '09: Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments, 2009.

[27] T. A. Majchrzak, A. Jakubiec, M. Lablans and F. Ückert, “Towards better social integration through mobile web 2.0 ambient assisted living devices,” in SAC '11: Proceedings of the 2011 ACM Symposium on Applied Computing, 2011, pp. 821-822.

[28] E. Aarts and J. Encarnacao, “True Visions. The Emergence of Ambient Intelligence”. ISBN 978-3-540-28972-2, Springer, 2008.

[29] A. Valli, “The design of natural interaction. Multimedia Tools and Applications,” vol. 38, no. 3, pp. 295-305, 2008.

[30] E. Aarts and B. de Ruyter, “New research perspectives on Ambient Intelligence,” Journal of Ambient Intelligence and Smart Environments., vol. 1, no. 1, pp. 5-14, 2009.

[31] F. Zhou, H. Been-Lirn Duh and M. Billinghurst, “Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR,” ISMAR 2008, pp. 193-202, 2008.

[32] C. Ramos, “Ambient intelligence - a state of the art from artificial intelligence perspective,” In Proceedings of the aritficial intelligence 13th Portuguese conference on Progress in artificial intelligence (EPIA'07)," Berlin, Heidelberg, 2007, pp 285-295.

[33] M. Alcañiz and B. Rey, “New Technologies For Ambient Intelligence, IOS Press,” 2005. [Online]. Available: http://www.ambientintelligence.org.

[34] N. Streitz et al., “Smart artefacts as affordances for awareness in distributed teams in The disappearing computer, Norbert Streitz, Achilles Kameas, and Irene Mavrommati (Eds). Lecture Notes In Computer Science,” vol. 4500, 2007, pp. 3-29.

[35] H. Nicolau and J. Jorge, “Elderly text-entry performance on touchscreens,” in Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility - ASSETS '12, 2012, pp. 127-134.

[36] M. Kobayashi, A. Hiyama and T. Miura, “Elderly user evaluation of mobile touchscreen interactions,” in Human-Computer Interaction – INTERACT 2011 Lecture Notes in Computer Science, 2011, pp. 83-99.

[37] N. Hollinworth and F. Hwang, “Investigating familiar interactions to help older adults learn computer applications more easily,” in BCS-HCI '11 Proceedings of the 25th BCS Conference on Human-Computer Interaction, 2011, pp. 473-478.

[38] C. Wacharamanotham, J. Hurtmanns, A. Mertens, M. Kronenbuerger, C. Schlick and J. Borchers, “Evaluating swabbing: a touchscreen input method for elderly users with tremor,” in Proceedings CHI '11 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2011, pp. 623-626.

[39] C. Stößel and L. Blessing, “Mobile device interaction gestures for older users,” in Proceedings of the 6th Nordic Conference on Human-Computer Interaction Extending Boundaries - NordiCHI '10, 2010, pp. 793-796.

[40] C. Leonardi, A. Albertini, F. Pianesi and M. Zancanaro, “An exploratory study of a touch-based gestural interface for elderly,” in Proceedings of the 6th Nordic Conference on Human-Computer Interaction Extending Boundaries - NordiCHI '10, 2010, pp. 845-850.

[41] J. A. Jacko et al., “Effects of Multimodal Feedback on the Performance of Older Adults with Normal and Impaired Vision,” in Universal Access Theoretical Perspectives, Practice, and Experience Lecture Notes in Computer Science, 2003, pp. 3-22.

[42] M. D. Lezak, D. B. Howieson, E. D. Bigler, and D. Tranel, “Neuropsychological Assessment, ” fifth edition, Oxford University Press, New York, 2004

[43] Weiqin Chen, “Gesture-Based Applications for Elderly People,” in Human-Computer Interaction. Interaction Modalities and Techniques, 15th International Conference, HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, pp 186-195