Applying a Multimodal User Interface Development Framework on a Domestic Service Robot

Michalis Foukarakis
Institute of Computer Science, FORTH-ICS
Vassilika Vouton
Heraklion, Crete, Greece
+30 2810 391733
foukas@ics.forth.gr

Margherita Antona
Institute of Computer Science, FORTH-ICS
Vassilika Vouton
Heraklion, Crete, Greece
+30 2810 391743
antona@ics.forth.gr

Constantine Stephanidis
Institute of Computer Science, FORTH-ICS
Computer Science Department, University of Crete
Vassilika Vouton
Heraklion, Crete, Greece
+30 2810 391741
cs@ics.forth.gr

ABSTRACT
Quality of life and active aging of an ever-increasing number of elderly people is becoming an essential objective for today's societies. To this end, the domain of assistive robotics is continuously researching and developing new robotic solutions for assisted living. One of the research challenges in this scope is the design of appropriate user interfaces for the elderly. This paper discusses the use of a multimodal user interface (UI) development framework for developing elderly-friendly robotic applications in the scope of the EU funded project RAMCIP. The project aims to develop a novel home robotic assistant system for older adults with mild cognitive impairment (MCI) and at early stages of Alzheimer Disease (AD). The paper describes the UI framework, its application in RAMCIP and the initial experiences regarding the use of the framework gathered from the preliminary pilot trials of the project with actual patients.

CCS Concepts
• Human-centered computing~User interface toolkits • Computer systems organization~Robotics

Keywords
Multimodal user interface; service robot; user interface adaptation.

1. INTRODUCTION
The successful integration of robotic platforms in domestic environments is emerging as the result of multidisciplinary efforts from various scientific fields, ranging from computer vision to machine learning and information systems. The challenges of this vision have been so overwhelming that more focus has been given so far to achieving safe and reliable robot functioning than to the actual human-robot interaction between the platform and its users. However, as the field of robotics has matured over the last years, a focus shift from the hardware itself to Human-Robot Interaction (HRI) is becoming necessary.

To this end, the FIRMA framework [1] has been recently proposed to support the development of elderly-friendly multimodal applications in robotic environments. This paper discusses the use of the FIRMA framework to build and support the user interface and the system integration of a socially assistive robot developed in the context of the EU-funded project RAMCIP[1].

The RAMCIP project's objective is to research and develop a novel domestic service robot, with the aim to proactively and discreetly assist MCI and AD patients in their everyday life. The RAMCIP robotic platform offers high-level cognitive functions, driven through advanced human activity and home environment modelling and monitoring, enabling it to optimally decide when and how to assist its users. The robot, apart from touch-screen, speech and gestural modalities, also incorporates an augmented reality display, as well as an underlying empathic communication channel allowing it to sense user affect and moderate it. In addition, the platform is equipped with a sophisticated anthropomorphic hand, manipulated though novel grasping and dexterity algorithms and capable of grasping and manipulating a variety of objects in realistic user homes, supporting also safe handover.

The RAMCIP robot is being developed based on a set of use cases which have been elaborated taking into account the analysis of collected users' requirements and expectations, and by assessing the robot's functions, ergonomy, and Human-Robot interaction modalities [2]. The general use case categories have been further analyzed into sub use cases (SubUCs), in which specific usage scenarios are described, such as fall detection, bringing water, assisting with medication intake, etc. All the robot's functionalities, including Human-Robot communication which has been developed using the FIRMA framework, cooperate in order to ensure optimal robot operation and user satisfaction throughout the SubUCs.

This paper focusses on HRI in RAMCIP and is organized as follows. First, a brief description of the FIRMA framework's main features are presented, followed by its application and enhancement in the context of RAMCIP. Then, the framework's integration to the robot's system is described. Finally, the RAMCIP user interface application developed using the framework is presented, along with the preliminary experience stemming from the pilot trials with actual users that have been conducted using the first version of the robotic platform.

PETRA '17, June 21-23, 2017, Island of Rhodes, Greece

ACM ISBN 978-1-4503-5227-7/17/06...$15.00
http://dx.doi.org/10.1145/3056540.3076187

[1]http://www.ramcip-project.eu/ramcip/

2. THE FIRMA FRAMEWORK

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 [3]. However, at present developing such interfaces is a very demanding task mainly performed ad hoc, due to the lack of supporting tools.

The FIRMA framework [1] targets to support the development of multimodal, elderly friendly, interactive applications for assistive robots. FIRMA provides developers with the necessary technologies, tools and building blocks for creating easy to use elderly-friendly multimodal applications on robotic platforms. The framework facilitates the effective and efficient development of the supported user interfaces, thus simplifying to a great extent the developer's work, by inherently offering multiple modalities, modality integration and user interface adaptations. This section describes the basic components and functionalities of the FIRMA framework which are relevant in the context of RAMCIP.

2.1 ACTA for Modality Integration and Application State

FIRMA makes use of the ACTA language proposed in [4] and has extended its runtime mechanism to support the development of applications. ACTA is primarily used for describing finite state machines (FSM). A finite state machine consists of states, inputs, and transitions. Authors can use ACTA for scripting a smart game or application in general, by outlining a set of finite states and event-driven transitions (i.e., a transition from one state to another is triggered by an event or a message). Thus, ACTA facilitates event driven programming, in which the flow of the program is determined by events, event handlers, and asynchronous programming.

ACTA is used in the FIRMA framework to fulfil three different purposes. The first one is to enable modality integration and fusion by consolidating all the modality generated events at a higher level understood by the applications. In this context, the developer can either handle the result of each modality individually or take advantage of the consolidation mechanism which is able to map individual but equivalent modality events into one. For example, a) a "Yes" gesture, b) the word "Yes" and c) the touch of a button on the screen with the text "Yes" on it are considered equivalent events and can be mapped to a single "Positive" event which is understood by the system. On the other hand, if the developer wants to differentiate the functionalities between modalities, they can write an ACTA script that maps the corresponding events to different actions than the default.

The second use of ACTA is maintaining application states and supporting state transitions. Application screens are represented as states of a finite state machine and the language supports transitions between them. The developer can include this kind of information in the application's script file. Each state can also include a number of rules, statements or conditions that describe the functionalities that the current state possesses. Common types of statements are: "ShowScreen(ScreenName);", where a method of the same name is called with the appropriate parameter and shows a specific screen, "when(condition){…}" blocks, which execute the statements within only if the condition is met and finally "NextState = StateName;" commands that indicate state transition. An example of this type of script can be seen in Figure 1.

The third and final use of ACTA is to enable different kinds of user interface adaptations. The adaptation manager of the framework includes an ACTA script file that contains several "when" statements that dictate the values of a number of adaptation parameters. For example, when the adaptation manager of the framework is informed by a change in the room brightness, an internal parameter named *CurrentBrightnessLevel* takes a different value which the ACTA script detects, and subsequently activates the corresponding rule to adjust the screen brightness accordingly. An example script of this type is shown in Figure 2.

```
State "SUBUC_1.1_FAQ_1_STATE_1"
{
    ShowScreen("SUBUC_1.1_FAQ_1_STATE_1");
    when(ButtonPressed == "BUTTON_POSITIVE")
    {
        NextState = "SUBUC_1.1_FAQ_1_STATE_2";
    }
    when(ButtonPressed == "BUTTON_NEGATIVE" ||
         GestureRecognized == "HELP" ||
         Time == 120
        )
    {
        NextState = "SUBUC_1.1_FAQ_1_OBS_2";
    }
    when(Time == 30)
    {
        NotifyLouder(LastTextSpoken);
    }
    when(Time == 60)
    {
        NotifyEvenLouder(LastTextSpoken);
    }
}
```

Figure 1. ACTA script for a specific communication state during the RAMCIP SubUC "Fall Detection".

```
when(CurrentBrightnessLevel == "DARK")
{
    ColorScheme = "LightColors";
    SetBrightness(20);
}
when(CurrentBrightnessLevel == "NORMAL")
{
    ColorScheme = "DarkColors";
    SetBrightness(50);
}
when(CurrentBrightnessLevel == "LIGHT")
{
    ColorScheme = "DarkColors";
    SetBrightness(90);
}
```

Figure 2. ACTA script snippet that defines user interface adaptations related to environmental brightness.

2.2 Interaction Modalities

The initial set of supported modalities in FIRMA includes speech input and output, gesture input and touch-based graphical interaction.

Speech recognition allows users to interact with and control speech-enabled applications. Users can interact with a robot using their voice, in the form of simple vocal commands. This is a set of pre-agreed verbal instructions that can be recognized and understood by the system. They are encoded into speech recognition grammars. The framework offers acquisition and monitor of speech input, support of speech recognition grammars that produce both literal and semantic recognition results, capturing information from events generated by the speech recognition, and full configuration and management of the parameters of the provided speech recognition engine. The engine used for supporting this modality has been implemented using the Microsoft Speech Platform.

Speech synthesis is the computer-generated simulation of human speech. The modality support in the framework has been implemented using two similar methods. The first method uses the Microsoft Speech Platform which provides a highly efficient speech synthesis framework through its Microsoft.Speech namespace. While this engine performs very well, its functionalities are limited by the selection of available voices. The second method utilizes the System.Speech namespace provided by the .NET framework and supports a greater number of voices. Both methods offer adaptation parameters that allow modification of speech volume, speaking rate and pitch.

Gesture recognition in the context of robotics refers to the technology that allows a system to understand hand and arm gestures made by the users. Since many gesture recognition techniques and engines exist, the framework itself does not include the capability to recognize gestures. However, it is equipped to receive a number of common gesture interpretations (such as Yes, No, Cancel, Help, etc.) and incorporate the information in its adaptation mechanisms. Therefore, any gesture recognition engine that is able to send gesture events to the framework is compatible.

2.3 Adaptation

The framework's elements, components, library controls and high level dialogues can be optimized around users' needs and preferences. The framework applies concepts of design for all [5] and universal access to produce an adaptation mechanism that is able to change the appearance of the generated user interfaces based on the profile of each user.

The framework supports automatic user interface adaptation based on both off-line (user profile) and on-line (user interaction monitoring) knowledge. The generated interfaces are tailored to the end user based on both adaptability (off-line adaptation) and adaptivity [6] (on-line adaptation) concepts. The former ensures that the applications adapt to the preferences and limitations of the user so that he is presented with a full usable interface to interact, while the latter is based on monitoring the interaction history and adapting the applications to dynamically changing factors or inferences, e.g., the ambient noise or the difficulties that the users are facing with the current user interface instance. The ACTA rules engine is used at runtime to respond to environmental and context of use changes to the system variables, and infer facts to be used by a decision making component to activate or deactivate the appropriate components of the adaptive component hierarchies on which the user interfaces are generated. The above methodology ensures that the end-user will experience an interaction experience tailored to his individual user attributes and to the particular context of use.

Various adaptation strategies are supported [1], which can be classified according to the impact they have on the user interface: conservation, e.g., simple scaling of UI elements; rearrangement, e.g., changing the layout; simplification / magnification, same UI elements but with modified presentation; increase (also called progressive enhancement) / reduction (also called graceful degradation), in terms of UI elements [7].

2.4 ROS interface

A common communication and application framework used in robotics is ROS [8]. This framework enables creating robotics applications and handles many of the communication hassles between software and hardware using messages and services. The ROS framework has been designed to run primarily on Linux systems, therefore the FIRMA framework, which runs on Microsoft Windows, includes a library for establishing communication between the two operating systems in order to be able to communicate with the rest of the software components running inside the platform. This scheme is also used in RAMCIP.

2.5 Other features

FIRMA includes a number of other features such as inherent support for globalization and localization, a permission manager for managing which application is allowed access to the screen, as well as support for screen navigation and orchestration. These features have been extensively exploited in the development of the RAMCIP user interfaces.

3. FIRMA'S ENHANCEMENT AND APPLICATION IN RAMCIP

This section reports on the experience gained in applying the FIRMA framework in the development of the user interfaces for the RAMCIP robotic platform, as well as on the necessary enhancements.

First of all, the applications that would be supported by the framework were designed. For the first version of the platform and the initial set of SubUCs implemented for the project, two applications were required. The first one, called FAQ (Frequently Asked Questions), included the majority of the dialogues spoken and shown by the robot, which were usually questions, but also notifications. The second one, called Dialer, was responsible for the robot's external communication with caregivers, friends and relatives.

FIRMA provides a set of commonly used user interface dialogues and screens (such as option presenters, notification screens and binary decision dialogues), which were extensively used in the implementation of the RAMCIP applications. Additional visual components were also introduced. The ability to optionally enhance the dialogues with images was added, as well as a new loading screen to be shown when the robot is busy performing tasks. For the Dialer application, a screen offering information about the current status of the call and options for cancel and hang up was implemented. These visual components are fully adaptive and generic enough to serve as a permanent addition to the FIRMA framework for future use. In addition, a timing mechanism for all the dialogues has been implemented that allows time based events such as repetition of a dialogue with louder voice or sound, time outs for dismissing dialogues or other time-related functionalities.

In order to enhance the remote communication capabilities of the platform, two systems were considered for addition to the framework. The first one was to use a well-known application programming interface for Voice over IP that allowed traditional calls to relatives or friends. The second was to integrate the popular messaging and voice/video call program Skype to the framework. Both methods were implemented, added to the framework and considered for use in RAMCIP.

The framework's implemented ROS interface for communication between the user interface running in Windows and the ROS nodes running in Linux was modified to support RAMCIP's specific capabilities and receive input from its sensors and software components.

A novel feature in modern user-friendly robots is the incorporation of robot emotions, commonly represented as facial expressions.

RAMCIP includes a separate screen exclusively used for showing animated facial expressions. The FIRMA framework was extended in order to support access to the facial expressions screen from all the applications.

Finally, the robotic platform specifications allowed for a number of additional FIRMA features, such as extra sensors and screens. The above mentioned enhancements were implemented in order to successfully and efficiently employ the FIRMA framework in the RAMCIP platform, exploiting the openness and ease of use of the framework.

4. SYSTEM INTEGRATION

This section describes integration of the FIRMA framework in the RAMCIP robotic platform. To illustrate the role of FIRMA in the context of the complete RAMCIP software and hardware system, the RAMCIP's features relevant to the developed applications (referred to collectively as the "user interface application") are presented in the following subsections along with the way that the framework interacts with each one of them.

4.1 Robot Software Components

The developed robot user interface runs on a 12 inch tablet mounted on the front side of the robot platform (see Figure 3). Since the majority of the software necessary for normal robot operation runs inside the platform on a Linux operating system, the ROS framework is used for communication between the devices.

The user interface application uses the ROS bridge of the FIRMA framework for establishing communication with three different software components running inside the Linux platform, the assistance decision maker (ADM), the virtual user model (VUM) and the gesture recognition engine.

The following subsections describe the relation and communication between the user interface application and the rest of the RAMCIP software.

4.1.1 ADM Integration and UI dialogue flow

The assistance decision maker component is responsible for realizing the high-level cognitive functions of the robot and making decisions about the overall behaviour of the robot, e.g., deciding whether assistance is needed and infer the optimal way that assistance should be provided (robot action/manipulation or communication). To implement the robot behaviour patterns, the ADM makes use of finite state machine concepts where each state corresponds to one robot action such as navigation, grasping, object detection, etc. or a communication action. The user interface application is informed through a ROS message about the communication action and undertakes the task to provide the appropriate dialogues to the user according to the instructions from the ADM. The message includes information about the user interface application that is required (FAQ/Dialer) and a unique identifier that the applications need to initiate the communication.

Each application is equipped with its own set of FSMs that have been modeled as ACTA scripts and correspond to series of dialogues that are presented to the user in accordance to the currently running scenario. When the FSM has reached a final state, it means that the result of the communication with the user needs to be fed back to the ADM so that it can decide on the next robot action. A visual example of such an FSM can be shown in Figure 4. In this example, each circle represents an FSM state for which ACTA scripts dictate the functionality. This typically means that a dialogue will be presented to the user, possibly including speech output, text and images. The ACTA script for STATE1 in the figure below is shown in Figure 1.

Figure 3. The RAMCIP Robot Platform V1.

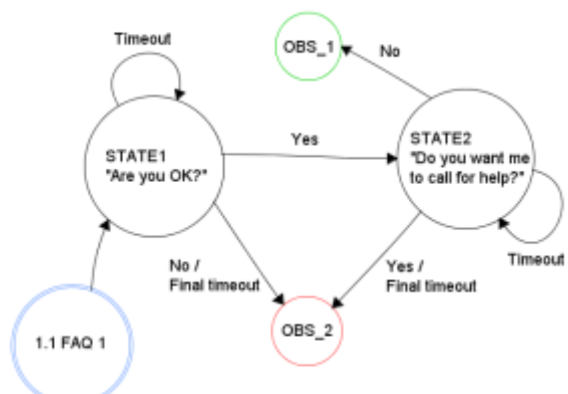

Figure 4. Communication FSM for part of the SubUC "Fall Detection".

4.1.2 VUM Integration

The virtual user model includes a number of parameters that describe the user, including skills and preferences. These parameters are either static (e.g., user name, communication preferences, etc.) or dynamic (e.g., mood). The user interface application retrieves all the static information at the beginning of operation to adapt its components accordingly. During operation, some dynamic parameters can be queried and used for online adaptations. Access to the VUM is provided by ROS.

4.1.3 Gesture Recognition Engine Integration
The gesture recognition engine used in RAMCIP [9][10] has been tested successfully in other projects and natively supports ROS integration. Therefore, it emits gesture messages which the user interface catches and translates into appropriate commands using ACTA scripts, as described in section 2.1.

4.2 Robot Sensors
While the robot contains several different sensors used for a variety of tasks, the user interface application makes use of two of them, the platform laser sensors and a light sensor. It also makes use of the array microphone's capability to detect environmental noise.

4.2.1 Laser Sensors
The RAMCIP platform includes laser sensors that can accurately detect the position of the user relative to the platform. This information is useful to the user interface application in many ways.

First of all, since FIRMA supports parameterized visual component sizing, distance information is used for determining the size to use for icons, buttons and text. The smaller sizes are easier on the eyes while interacting through the touch screen from close range, but in cases where the user is a bit further away from the screen, bigger letters are more convenient and legible. The user interface application constantly receives information about the user distance from the platform and adjusts the component sizing accordingly, falling back to a default size in case there is no distance information. This is used in order to compensate for potential sensor malfunction or inability to detect the user. Examples of component sizing are shown in Figure 5.

Figure 5. Different sizes and colors for the user interface.

Distance information can also be useful for modality selection, since not all modalities are efficient in all distances. Touch interaction is only useful at very close distance. Gesture recognition works best only if the user is between one and three meters away from the platform and it is required that the user's skeleton is visible and has been correctly detected by the front camera of the robot. Moreover, voice recognition has a maximum efficient range. Taking into account all the above, the input of the laser sensors which is provided through a ROS service that provides the distance value can be used as one of the parameters that determine interaction modality selection and activation/deactivation.

Finally, the dialogues that are presented to the user through the touch screen of the tablet can be further adapted based on knowledge of the distance between the user and the robot. In some cases where the user is not close to the robot, a combination of an image and speech synthesis in place of text could be preferable to convey the necessary information to the user.

4.2.2 Light Sensor
The robot is equipped with a simple light sensor that can measure light intensities up to 1000 lux. The sensor is directly connected to the tablet. These values are then translated by the user interface application into three different states of illumination (dark, normal, light) which are then used in an ACTA script to adjust the tablet's brightness (see Figure 2).

4.2.3 Noise Sensor
The robot's microphone array is able to determine the ambient noise from the environment. This information is used by the user interface application to turn on or off the speech recognition modality, according to an ambient noise level threshold.

4.3 Robot Hardware
The user interface application interacts with two important hardware components of the RAMCIP platform, the augmented reality projector and the facial expression screen.

4.3.1 Augmented Reality Projector
RAMCIP's head integrates an Augmented Reality Interface for projection of information over the environment. Using an RGB-D sensor and a LED video projector, the interface allows capturing 3D information of the surrounding area and thanks to the Degrees of Freedom (DoF) of the head, it can be used to identify objects in the scene or reconstruct the environment. This sensor, coupled with the projector, includes the capability for the robot to point to specific objects or part of them by means of light projection over object surfaces. The system allows the robot to identify an object and project over that an image that matches the geometry of the object. This allows the creation of object highlight as a torch light, or alternatively the projection over a planar surface of a generic image.

The FIRMA framework is able to communicate with the software controlling the projection interface using specific commands such as "highlight the pillbox" through messages sent using the ROS bridge. The projection capability has been added to the arsenal of supported interaction modalities and is integrated in the adaptation plan of RAMCIP.

4.3.2 Facial Expression Screen
An additional screen has been mounted on the robot head and is used exclusively for showing the facial expressions of the robot. Its content is managed by the user interface application, which has defined a set of rules that define the facial expression to use for each dialogue and robot state. The transitions between emotional states of the robot face have been modeled using intermediate images which are shown in order to provide a sense of animation. Facial expressions include the following emotional states: neutral/calm, excited, sad, tired, sleeping, focused and angry. The screen is in essence a second screen connected to the tablet and consists of a simple borderless window that is created during the user interface initialization and constantly shows the current facial expression of the robot as an image.

5. PRELIMINARY PILOT TRIAL EXPERIENCE
According to the RAMCIP's implementation and evaluation plan, two phases of pilot trials corresponding to two versions of the platform to be manufactured are planned. In each phase, a subset of the original use case set is tested. In the first phase, which has been very recently completed, only the most critical SubUCs (e.g., fall detection, medication uptake monitoring) have been tested, while in the second phase a more complete set of SubUCs will be tested.

The first version of the robotic platform was tested in a controlled real-home simulated environment with both healthy (7 male/3 female) and target user (5 male/3 female) patients. The controlled environment used for this phase of pilot trials was a hospital room in Lublin, Poland that was decorated and equipped with a sofa, a kitchen unit, a sink, a fridge and a table with two chairs so as to resemble an apartment. The main goal of the preliminary trials with healthy volunteers and MCI patients was to determine any specific technical difficulties that may occur in a real elderly persons' home environment as opposed to the lab environment.

For each user, the first planned set of SubUCs was tested by performing the planned SubUC scenarios in order. For example, first the "fall detection" scenario was simulated, then in the next scenario the user dropped an object on the floor which the robot detected and tried to pick up and so on. As regards the performance of the modified FIRMA framework during the first pilot trials, the following observations hold:

- **User interface look and feel**: The users liked the initial set of offered themes and sizes and preferred the dark blue theme with small button sizes. This was set as the default look and feel configuration throughout the pilot trial period.
- **Communication with relatives**: The external communication functionality performed adequately and there were no serious issues conducting voice calls with other persons. The voice call platform that was selected to be used in the trials was Skype due to its stability and convenience. A common remark was that the Skype integration in the user interface could be a bit more automated and streamlined. Another concern was that in real environments most persons own a telephone number but not everyone owns a Skype account and those that do are not always online.
- **Speech Synthesis**: The free Polish voice Paulina offered by Microsoft was used as RAMCIP's text-to-speech voice. The users had some difficulties understanding some phrases, especially when the speaking rate of the voice was on the high side. This could be attributed to 1) the quality of the used voice, 2) the fact that some of the users had hearing impairments or were old enough to have lower than average hearing and 3) the speaking rate setting of the voice which could be higher than the optimal setting considering the target user group.
- **Speech Recognition**: Initially, there were issues with the speech recognizer. Sometimes it would not recognize the user commands and other times it would falsely detect commands. The most critical failure cases were solved by configuring the array microphone of the robot and by adjusting the speech recognition threshold accordingly. After these modifications, the speech recognizer performed more smoothly. It was important to note that this was the users' preferred modality over touch screen and gestures.
- **Adaptations**: The objective of the first pilot trials in the controlled lab environment was to primarily evaluate the core RAMCIP functionalities including navigation, grasping and communication. During the pilot trial period either some of the adaptations were not available or there was no time or opportunity to properly evaluate them. Therefore, the user interface adaptations that were based on the user preferences and the ambient noise detection were the ones that were enabled during the period.
- **System Integration and Stability**: The FIRMA framework was integrated with the rest of the system and the communication between the Windows application developed using it and the Linux machine inside the RAMCIP platform was successful at all times. The only problems encountered during the trials were hardware related or the failing software was running in the Linux platform. In all these cases, a simple restart of the Windows application would reestablish the communication flow without issues.

At the time of writing, the second version of the robotic platform is being manufactured, and will be followed by the second phase of pilot trials in user homes. Taking into account the experience from the first preliminary testing period, a number of improvements to the user interface application have been considered for inclusion in the second version of the platform.

Firstly, regarding the communication with relatives/friends, the Voice over IP system can be investigated as a viable alternative to Skype, to alleviate reliance on a third party application. However, this method will incur additional costs to the users as it is more expensive. Secondly, the speech synthesis issues can be solved either by using a higher-quality commercial voice or by carefully selecting the appropriate intonation parameters and especially the speaking rate. Thirdly, speech recognition can be improved by having a dynamic recognition threshold that is adjusted differently according to context (e.g., simple Yes/No commands are more easily recognized than more complex phrases) and by configuring the array microphone properly as observed during the first pilot trials. Finally, extensive lab testing using the adaptation and modality fusion capabilities of FIRMA will need to be performed.

6. CONCLUSION

This paper has presented the application of the FIRMA framework in the context of the socially assistive RAMCIP robot. Overall, it can be concluded that the FIRMA framework has greatly facilitated the development of the RAMCIP platform applications with respect to the ad hoc development of the various modalities and their integration from scratch. The framework has also significantly eased the overall integration of the applications in the platform through ROS, as well as the provision of user interface adaptation and multilinguality. While the framework already supported many of the features that developers require for developing elderly-friendly multimodal robotic applications, its core has been also extended and improved for implementing the multimodal adaptive user interface of RAMCIP, exploiting the openness and ease of use of the framework.

The first pilot trials with the integrated platform conducted in the scope of RAMCIP provided useful feedback for further improving the user interface and the FIRMA framework to be used in the second round of pilot trials. Further work in this direction is anticipated to focus on improving the capabilities and performance of the speech synthesis and speech recognition modalities, as well as on upgrading the adaptation manager component with concrete adaptation rules based on a more detailed user profile and including additional context of use adaptations.

7. ACKNOWLEDGMENTS

This work has been supported by the EU Horizon 2020 funded project "Robotic Assistant for MCI Patients at home (RAMCIP)" under the grant agreement with no. 643433.

8. REFERENCES

[1] Kazepis, N., Antona, M., and Stephanidis, C. 2016. FIRMA: A Development Framework for Elderly-Friendly Interactive Multimodal Applications for Assistive Robots. *The Ninth International Conference on Advances in Computer-Human Interactions* (April. 2016), 386-397.

[2] Gerłowska, J., Abdelnour, C., Grabowska, K., Hernández, J., Korchut, A., Martín, E., Rejdak, K., Skrobas, U., Szczęśniak-Stańczyk, D., Szklener, S., Tantinyá, N. 2015. Report on end user requirements and use cases break down. RAMCIP Project Deliverable, http://www.ramcip-project.eu/ramcip/sites/default/files/documents/ramcip_deliverable_d2.1_1.pdf

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

[4] Zidianakis, E. 2015. Supporting Young Children in Ambient Intelligence Environments, Ph.d. Thesis, University of Crete, Computer Science Department, Heraklion, Crete, Greece.

[5] Stephanidis, C., Salvendy, G., Toward an Information Society for All: An International R&D Agenda, *International Journal of Human-Computer Interaction*, vol. 10, no. 2, 107-134, 1998.

[6] Savidis, A., Stephanidis, C., Unified User Interface Design: Designing Universally Accessible Interactions, *International Journal of Interacting with Computers,* 16 (2): 243-270, 2004. DOI=http://dx.doi.org/10.1016/j.intcom.2003.12.003

[7] Paterno,F., User Interface Design Adaptation, *The Encyclopedia of Human-Computer Interaction*, 2nd Ed., 2013.

[8] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In *ICRA workshop on open source software* (Vol. 3, No. 3.2, p. 5).

[9] Michel, D., Papoutsakis, K., Argyros, A. Gesture recognition for the perceptual support of assistive robots, *International Symposium on Visual Computing (ISVC 2014)*, Las Vegas, Nevada, USA, 2014

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