AdaptO
Adaptive Multimodal Output

António Teixeira, Carlos Pereira, Miguel Oliveira e Silva, Osvaldo Pacheco, António Neves
DETI/IEETA, University of Aveiro, Campus Universitário de Santiago, Aveiro, Portugal

José Casimiro
Polytechnic Institute of Tomar, Tomar, Portugal

Keywords: HCI, Multimodal output, Context-awareness, Universal access, User, Intelligent agents, Speech synthesis, e-Health, Distributed systems, Smart environments.

Abstract: Currently, most multimodal output mechanisms use a very centralized architecture in which the various output modalities are completely devoid of any autonomy. Our proposal, AdaptO, uses an alternative approach, proving output modalities with the capacity to make decisions, thus collaborating with the fission output mechanism towards a more effective, modular, extensible and decentralized solution. In addition, our aim is to provide the mechanisms for a highly adaptable and intelligent multimodal output system, able to adapt itself to changing environment conditions (light, noise, distance, etc.) and to its users needs, limitations and personal choices.

1 INTRODUCTION

Multimodal interaction aims to improve the accessibility of content. It allows an integrated use of various forms of interaction simultaneously(sound, gesture, GUI, etc.). It also intends to create an environment where a user accesses, transparently, the same content, regardless of the device (mobile phone, PDA, computer, etc.).

Multimodality allows a user to interact with a computer by using his or her own natural communication modalities, such as speech, touch and gestures, as in human-to-human communication. Multimodal interaction constitutes a key technology for intelligent user interfaces (IUI).

Initially the interaction had to be adapted to a given application and for a specific interaction context. The present diversity of environments, systems and user profiles leads to a need for contextualization of the interaction. "Nowadays, the interaction has to be adapted to different situations and to a context in constant evolution" (Rousseau et al., 2005a).

This diversity of the interaction context emphasizes the complexity of a multimodal system design. It requires the adaptation of the design process and, more precisely, the implementation of a new generation of user interface tools. These tools should help the designer and the system to make choices on the interaction techniques to use in a given context (Rousseau et al., 2005a).

Users have individual differences due to anatomical and physiological factors (eg, gender, brain dominance, vision, hearing and mobility/motor skills), psychological factors which are difficult to categorize and quantify (eg, processes and cognitive styles, skills, motivation, attention and concentration) and cultural or environmental factors (eg, language, ethnicity or culture). Efficient multimodal interfaces should also be able to take into account user's requirements and needs. Fast automatic adaptation to the user characteristics (ex: hearing abilities) is very important for usable multimodal systems (Karpov et al., 2008).

As stated in (Dumas et al., 2008), "less research has been done on multimodal fission than on fusion. Most applications use few output modalities and, consequently, employ straightforward fission mechanisms." But the output modalities have the very important role of transmitting information to the user. The relevance of improved fission and output is particularly relevant for groups of persons with some of their senses affected, such as older adults, or com-

Teixeira A., Pereira C., Oliveira e Silva M., Pacheco O., Neves A. and Casimiro J. (2011).
AdaptO - Adaptive Multimodal Output.
In *Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems*, pages 91-100
DOI: 10.5220/0003372500910100

pletely compromised (ex: deaf people). There is a need to create multimodal interaction with a more balanced use and adaptation to user and context of input and output modalities.

On the other hand, from the programmer and the application builder point of view, modular and simple multimodal programming toolkits are required in order to ease their adoption and to extend their use.

In this paper we present work on the output adaptation to users and context of an agent based multimodal system. Our aim is to provide the mechanisms for a highly adaptable and intelligent multimodal output system, able to adapt itself to changing environment conditions (light, noise, distance, etc.) and to its users needs, limitations and personal choices. This multimodal system is aimed at being an essential part of a new telerehabilitation system in development as part of the Living Usability Lab (LUL) Project (www.livinglab.pt), presently at the prototype stage (Teixeira et al., 2011). LUL is a Portuguese industry-academia collaborative R&D project, active in the field of live usability testing, focusing on the development of technologies and services to support healthy, productive and active citizens. The project adopts the principles of universal design and natural user interfaces (speech, gesture) making use of the benefits of next generation networks and distributed computing.

The paper is structured as follows: section 2 presents the concept of a new telerehabilitation service. Section 3 briefly presents, the most relevant related work. In section 4 we present the requirements within our objectives for AdaptO. Section 5 presents our proposal for adaptive multimodal output, including architecture, adopted technologies and information on the modules developed as proof of concept. Section 6 shows some example scenarios of AdaptO within the telerehabilitation service. Finally, sections 7 and 8 present a critical view of our work, showing what is not yet done while drawing some conclusions and future research.

2 A NEW TELEREHABILITATION SERVICE FOR THE ELDERLY

In very general terms, the new telerehabilitation service in development (Teixeira et al., 2011) aims at providing supervised remote exercise sessions at home or community centres, as a mean to maintaining health and prevent illness. Table 1 presents some additional information on the service.

The creation of a prototype for the service depends on, amongst other things, the development of two applications with suitable Human Computer Interaction, one for the elderly at home, other for the health professional planning, monitoring and evaluating the session (Fig. 1). The two applications use multimodal input and output, with particular emphasis in the use of speech and text. The use of speech derives from its potential to be usable by visually disabled people and to enable interaction hands free and at some distance from the devices. This capability of receiving information and giving commands to a computer at a couple of meters of the TV/computer display is essential when the aim is making all body movement exercises.

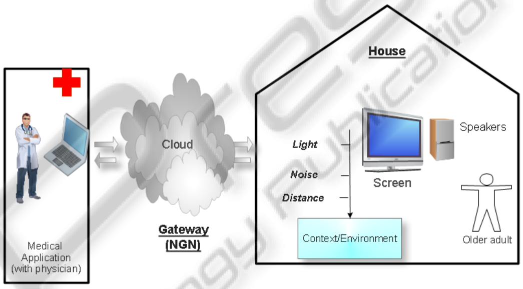

Figure 1: General view of the Telerehabilitation service, testbed for the multimodal output prototype presented in this paper.

3 RELATED WORK

The most relevant work addressing more directly the problem of adequate and adaptive multimodal output are (Rousseau et al., 2005b; Rousseau et al., 2005a; Rousseau et al., 2006; Coetzee et al., 2009).

Rousseau and coworkers (Rousseau et al., 2004; Rousseau et al., 2005b), in 2004, proposed the WWHT model, dividing the life cycle of a multimodal presentation in four steps:

1. What is the information to present?
2. Which modality(ies) should be used?
3. How to present the information using the modality(ies) selected?
4. and, Then, how to deal with the output results?

The first step, called the “semantic fission”, splits the information provided by the dialog manager into elementary information. The second step allocates a modality for each piece of information. The third step instantiates the selected modality with the information. These three steps are related to the construction of a multimodal presentation. The last step

Table 1: LUL Telerehabilitation Service with Multimodal Interaction.

Description	Remote supervised exercise sessions at home or community centres, for maintaining health and prevent illness. Sessions carried out concurrently at several sites via networked multimodal applications. A health professional supervises everything from the training centre/hospital, including the biosensors signals captured remotely and the (multiple) cameras images. The system also includes mechanisms to request and process information regarding effort level from the user.
End user	People taking rehabilitation sessions at home and people wanting to do exercise in case it is just training sessions.
Stakeholders and Roles	Health/Sport professionals (physiotherapists, gerontologists, etc.) who configures the sessions and directs them. Healthcare services provider, which should install and maintain the platform.
Technological description	The main user interface is a large size computer monitor (acting as a large size TV) combined with speakers and video cameras. In addition, it should be possible to use a set of biosensors. Sensors gather the vital signals from the patient and send them to the health professional.

(Then), renders the multimodal presentation to users (Rousseau et al., 2005a).

The same researchers also developed the MOSTe tool (Rousseau et al., 2005a; Rousseau et al., 2005c) which is intended to ease the specification of multimodal interfaces, realizing the first step of the WWHT model. It is composed by four editors (component editor, context editor, information editor and behavior editor). They are all graphic tools and express the behavior of the system by a set of rules. Each rule is called an election rule and is based in the if-then instruction. Each one belongs to one of three types: contextual, composition and property. The architectural model design at MOSTe is exported in XML format and implemented by the MOST system (Multimodal Output Simulation Tool). This second tool is responsible for selecting the correct output modalities, realizing the second and third steps of the WWHT model. Figure 2 shows the MOST architecture.

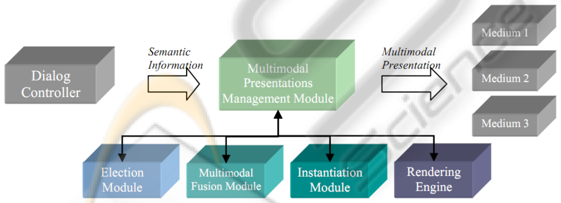

Figure 2: The MOST architecture, from (Rousseau et al., 2005a).

Coetzee, Viviers and Barnard (Coetzee et al., 2009) proposed a model to determine the best possible combination of input and output modalities. This model, designed Cost Model, is a mathematical tool that takes in consideration the user's profile and preferences. The user profile is defined in terms of abilities rather than disabilities. Preferences means how the user uses his five senses to interact with the system, and his literacy. The Cost Model takes in consideration these factors, and defines a set of vectors to determine the best solution for a specific user:

- a vector p, of real values scaled between 0 and 1 of length n, where each element in the vector represents the user's abilities: can talk, can click, can move pointer, can read, etc.;
- a vector w, which represents an adjusted user profile as based on his perceptual preferences. This vector is defined by multiplying the S matrix by the transposed vector p, where S is a diagonal matrix of size n x n, containing real values scaled between 0 and 1 to represent the perceptual user preferences;
- a vector c, which represents a cost estimation of suggested components to be used per adjusted user. The cost vector is defined by multiplying the D matrix by the transposed vector w, where D is a matrix of size nxm (n is the number of modeled user abilities and m the number of modeled available input and output components). Matrix D represents the relation between the user abilities and the required modalities. A larger value in c indicates which are the most important components for a specific user profile.

4 REQUIREMENTS

From the identified problems and aims for our multimodal system, several requirements were identified:

- Adaptability of the interaction according to the user abilities, allowing equivalent utilizations possibly by means of different multimodal interactions.
- Information on user capabilities, preferences and eventual historic usage information must be available;
- Information on context (ex: environment conditions) must be available;

- Redundancy of modalities must be used in order to increase the chances of message delivery ;
- Based on the environment and user, output modalities must have capabilities to decide to activate/deactivate themselves. Ex: there is no reason to keep active a TTS output when the user is deaf;
- Implementation of a registry upon which modalities may inscribe themselves by category, simplifying the search for possible modalities by the system.
- Inclusion of Speech output (synthesized or not) to enable use of the system by vision impaired persons;
- Use of several output modalities to enable use of the system by speech and hearing impaired persons;
- Output characteristics, such as the volume of the synthesized speech, must adjust themselves according to user's and environment (ex: distance to speakers, noise level and users hearing acuity);
- Speech rate must be adapted to listener and listening conditions (ex: distance, hearing abilities);
- If possible, allow users to be informed through their preferred modality(ies);
- The system should be modular, that is, the future inclusion of modalities should be as easy as possible, and without making changes to the core of the system.
- Fault-tolerance and extendibility should be taken into account on the architecture design.

5 ADAPTIVE MULTIMODAL OUTPUT (ADAPTO)

5.1 Rationale and Architecture

According to (Dumas et al., 2009a), "the generic components for handling of multimodal integration are: a fusion engine, a fission module, a dialog manager and a context manager, which all together form what is called the integration committee" (Fig. 3).

However, this architectural scheme implicitly assumes that input and output devices are simple dummy devices responsible only for sending input information to the system and to receive output messages already adapted to the context and user. In this approach the fusion and fission coordination services, are required to be very knowledgeable of all the available input and output devices, making it potentially very complex and more difficult to scale and extend our applications with new input and outputs devices. This problem is particularly important in the output devices since they convey the information to the users, closing the interaction loop with the application.

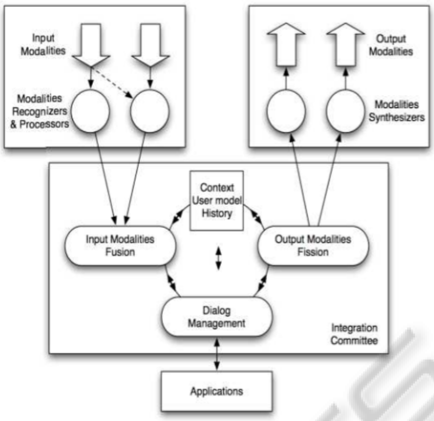

Figure 3: Architecture of a Multimodal System (Dumas et al., 2009b).

An alternative approach (Fig. 4), is to enhance the intelligence of output devices, making them able to adapt themselves to a dynamic context and to the user. In this way the responsibility to ensure adaptability in the applications is not centralized in a potentially very complex and knowledgeable fission engine, but is shared with the output devices themselves.

A higher degree of adaptability is achieved by separating two different aspects: a varying context environment, and specific knowledge of each application user. In the context adaptation the output interface of the application may adapt itself to varying light, noise conditions, the distance of the user, and other environment conditions. When user aspects are taken into consideration, output agents may adapt themselves to different users, such as a speech synthesizer to become disabled in the presence of a deaf user, or due to expressed interface preferences by the user.

To show why such architectural choice might be preferable, let's consider the case of an application using a speech output device. In a centralized approach, the fission engine would be required to know that a noisy environment, or a deaf user, may invalidate this type of output, and act accordingly. In our proposal, the speech output device handler, would be the one required to have such knowledge, and the one responsible the notify the (simpler) fission engine that it could not send the message to the user.

Considering that there could exist many different output devices in a single application, the impact of this architectural option on the system's complexity is expected to be very significant.

A collateral desirable side effect of this approach

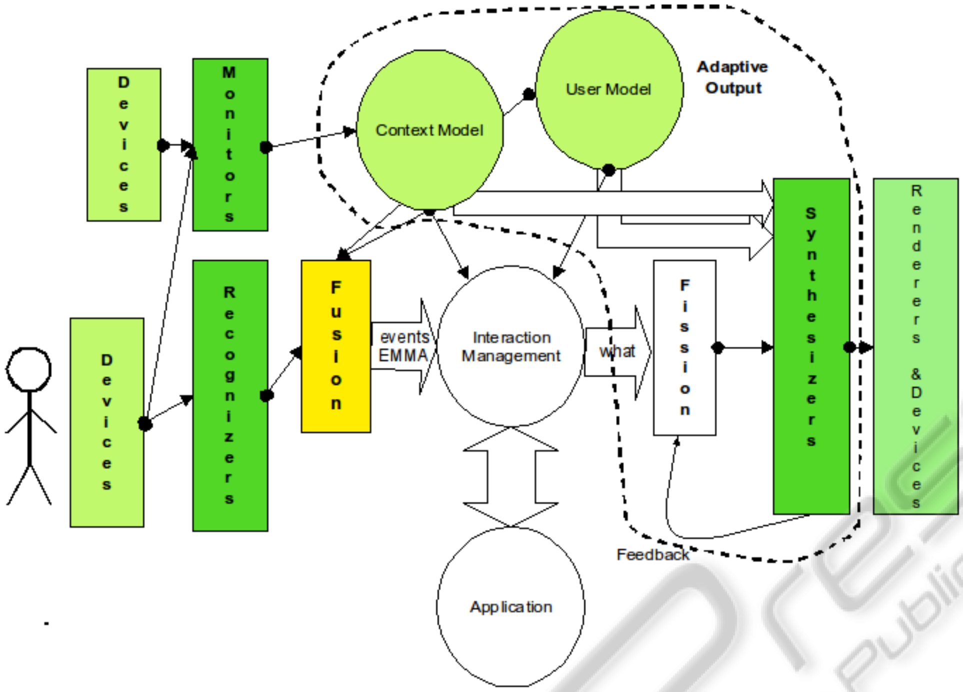

Figure 4: AdaptO Architecture.

is the possibility to have a more abstract fission engine. Output messages can became more abstract, making it easier to implement fault tolerant redundant output in applications, and also providing a simpler programming context for our dialog manager. For example, a textual message can be output either by a text message graphical window, or by a speech synthesizer. From the application point of view such detail is not important (as long as the message is successfully transmitted), so the dialog manager should only be knowledgeable of this more abstract type of output message, delegating concrete output realizations of it (text window or speech, in the example) to the availability of autonomous output agents.

The fission engine would be responsible for reporting either a successful transmission of the message, or its failure. In both situations, however, the specific output devices involved are abstracted away from the application.

5.2 Technologies Adopted: SOA and Agents

A natural choice to achieve a more autonomous and intelligent behavior in the output devices is to make them JADE agents (JADE, 2010). The choice of using an agent based platform is one of the best available options, because it is a mature and quite stable technology, able to provide us also with a solution for a distributed heterogeneous system, supported on a standard communication protocol (FIPA, 2010), which may simplify future integration of third party services supporting new input and output devices. Agents also provided us a versatile solution for the need of a decentralized, scalable, adaptable, and intelligent system.

Agents were also used to implement the other services in our architecture: the input devices, the context, user and history engines; and the fusion, fission and dialog manager services.

To reduce the cohesion between all these services, an event based communication scheme is used to simplify and abstract away the knowledge required for the system to operate. An output agent required to know about the noise level of the environment, or the user distance to the speakers, registers itself in the context agent to receive this type of information, decoupling itself from specific input agents able to extract such information. It may also register itself in the user model service to become aware of possible hearing or comprehension user problems. With such knowledge, this device may change the volume of the sound, and the rate of the speech to maximize a successful transmission of the messages.

All available output agents register themselves in the fission agent for output message types they are able to transmit, thus making the fission agent knowledgeable of the available (abstract) output agents, hence able to ensure fault tolerance.

There are also two more important services (also implemented as agents): a logger service able to register all the relevant history information for latter use, such as for the creation of user specific information for the user model. A coordination fission service able to ensure the reliability of the application as a whole. This service would be able to ensure fault tolerance, notifying the application when an output was not transmitted (for example, due to the unavailability of proper output devices).

5.3 Environment Monitoring Agents

At the time of writing, three environment monitoring agents are available: an environment background "noise" level estimator; a light conditions estimator; and a distance of the user to the screen/display estimator.

The "noise" level is obtained by using an event of the speech recognizer, `AudioLevelUpdated`. Based on a given variable, the algorithm records a number of values from the environment and makes an average between them. Given the capabilities of the sound recognizer, the interval defined is normally of 1 or 2 seconds.

The light conditions are evaluated based on statistical measures – Mean Sample Value (MSV) – of the intensities histogram calculated on the acquired image.

The distance is obtained using algorithms based on background subtraction to estimate the position of the person in the image. Using the properties of the vision system (position, camera and lens properties, etc.), it is possible to estimate the position of the person related to the camera.

5.4 Context and User models

The environment monitoring agents are all in communication with the **context model**. As mentioned earlier, this service, implemented as an agent, is responsible to register, in real-time, all the relevant environment conditions such as noise, light, distance of the user to the output devices, etc.. The information for this service comes from a set of specialized producer agents that, in general, are connected, directly or indirectly, to sensors, such as microphones and cameras.

When a change happens, for instance, on the distance agent, he alerts the context model which will (if it perceives as necessary) alert other agents when the distance parameter may influence their execution state (a text synthesizer for instances). As such, the context model functions as a gateway between environment agents and application-derived agents.

A **user model service** is also provided, in order to register and fetch specific user related capabilities and preferences. Examples of capabilities are vision and hearing acuity, and mobility capabilities. As preferences, we can have, for example, the personal preference for receiving information visually or for a specific color for text. In truth, this model may be used for two things: one, to disambiguate possible indecisions (on fusion module); two, improve the comfort and usability for users when interacting with the system.

The user model differs from the context model in that it can be used as a service. As such, the user model was design as web service, allowing for several systems to use its content. With it, it is our intention, that in the future, statistical information may be derived from the data enabling us to analyze the user's preferences and thus improving the system.

5.5 Synthesizers

The agents capable of transmitting information from the system to the user – named synthesizers in multimodal interaction literature – include two important mechanisms: The first is capable of deciding if in the current environment conditions and taking in consideration user capabilities it is in position to be active and fulfill the request. The second changes how the message is rendered, also based on contextual and user information 5.

The text Output and Text-to-Speech capabilities of the system were made available by a speech synthesizer agent and a text synthesizer agent, the first implemented using Microsoft's Speech Platform (Microsoft, 2010) and the other using Java Swing.

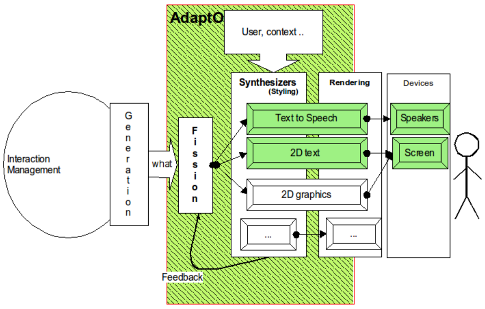

Figure 5: AdaptO output.

Presently, and as proof of concept, the text synthesizer varies, using simple heuristics, the font size as a function of the user visual capacity, environment lighting conditions and distance of the user to the screen.

In general the process of adaptation of a parameter from a synthesizer consists of 3 steps (Fig. 6):

1. calculation of individual gains/multiplication factors, k_i, based on the factors chosen to affect the parameter;
2. combination of the gains to obtain an unique gain, $K = \prod_i k_i$;
3. transformation of K into the range accepted by the renderer/device.

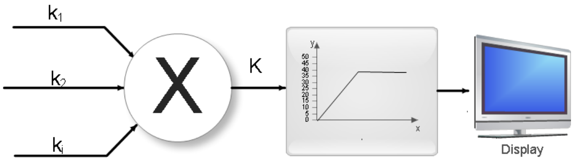

Figure 6: Parameter adaptation process.

Distance is measured according to the meters that separate the user from the output mechanism (for instance, a monitor). Anything that exceeds a predefined limit (10 meters in our first prototype) is ignored by the system. The output agent synthesizers that are dependent on the distance shutdown when the values exceed this limit.

Vision acuity is in our first approach an "abstract" value, from 0 to 10, where 0 represents a blind person and 10 a person with perfect vision. We expect in the near future to replace this by the use of the real results from medical visual evaluation. Vision capabilities below a threshold imply the shutdown of the synthesizer dependent on this factor, thus giving preference to other output modalities.

Text size presently depends on two factors: distance of the user to output device and the vision capabilities of the user. This variation is shown by Figure 7.

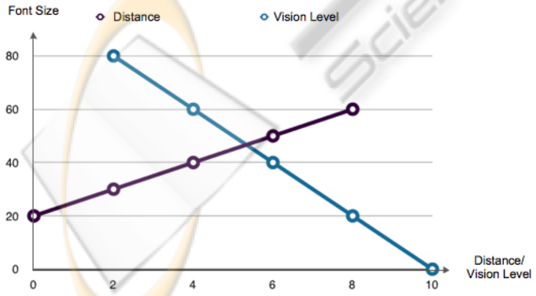

Figure 7: Text size variance according to the user's distance and vision capabilities.

The speech synthesizer is capable of varying both volume and speech rate. This increases the chances that the message is received and understood. The context and user models are crucial to make these two mechanisms possible. TTS volume uses a similar list of factors, with the light information replaced by environmental noise information. For now the adaptation of the TTS speech rate is only based on the user age, with k_i decreasing linearly starting at 40 years of age (based on the results reported in (Gordon-Salant, 2005)).

At the time of writing, the scales and the formulas used are quite simple. We used linear functions in order to test our ideas, but in the future we expect to replace them by more realistic ones.

As stated before, contrary to other systems, the synthesizers on our model are not mere output providers. They analyze their environment. They are capable of calculating critical parameters for their functioning and determine whether they can or not satisfy output requests.

6 FIRST RESULTS

As already mentioned, the testbed for AdaptO is a new telerehabilitation service for the elderly. In this section, we present some possible runtime scenarios that occur in this application as well as some preliminary results regarding output adaptation to context and user constraints.

6.1 Examples of the Adaptation Mechanism in Action

Serving as illustrative examples and also to better explain the system described above, three scenarios are presented with different user and context adaptations.

6.1.1 Scenario 1: Choice of One Synthesizer based on the User's Preferences

The system intends to output a message to the user and the user is close to the screen. The system reads the modality-registry and gets the information that both the text and the speech synthesizer are ready to output the message. However, the system also knows, by consulting the user model, that the user prefers to be notified via text messages and as such the message is transmitted via text. The sequence of actions and the intervening agents are illustrated in Fig. 8.

6.1.2 Scenario 2: Synthesizer Becomes Unavailable Due to Context

The system needs to transmit another message to the user. Assume that initially, both synthesizers were available and registered in the modality registry.

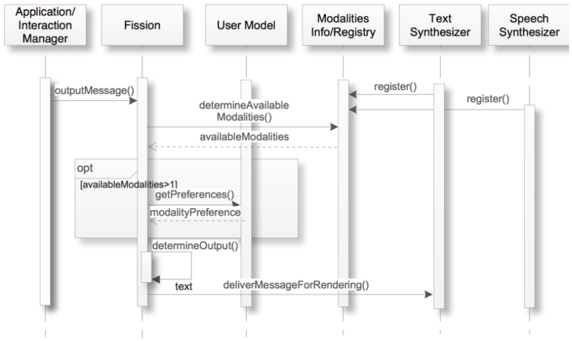

Figure 8: Scenario 1.

Nonetheless it was detected that the user went beyond the text synthesizer's range. When this happens, and since the text synthesizer is dependent from distance parameters, the context model alerts the text synthesizer to this fact. The text synthesizer recalculates its parameters and sees that it cannot output messages in these conditions and as such notifies the modality registry that he is offline. So, the system only has an option to provide the message and outputs it via speech. Fig. 9 provides detailed information on the actions and agents involved.

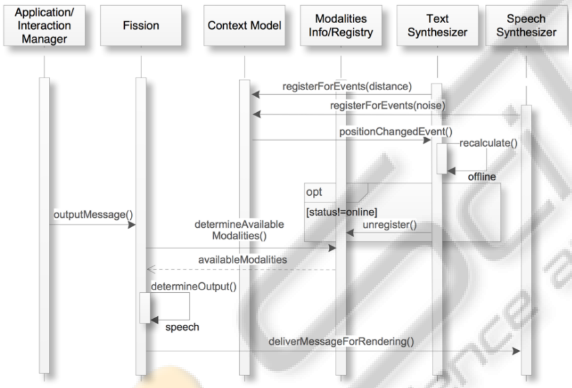

Figure 9: Scenario 2.

6.1.3 Scenario 3: Synthesizer Adjusts its Parameters Due to Context

Another message is available to be transmitted to the user. The user became once again in range of the text output, but, this time, a high level of background noise is present in the room. As such, following the same pattern on previous scenario, the speech synthesizer disconnects himself. The text synthesizer however is online. Since the distance changed, the font size also was recalculated proportionately. As the user is in range, the system outputs the message.

These three scenarios illustrates the adaptive nature of our proposal. In our view, the system gains in usability and capability besides becoming more fault-tolerant, critical in real-time systems.

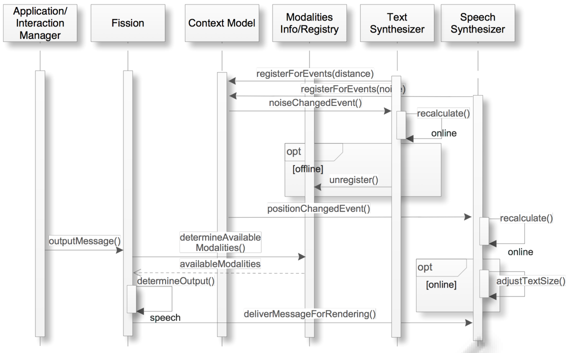

Figure 10: Scenario 3.

6.2 Output Examples

To provide some information of the actual results obtained with the present prototype, we present examples of the output created by the text synthesizer - the one better suited for inclusion in a written document as this paper - in response to combinations of groups of factors. The combined effect of the user's distance to the output device and the vision capabilities of the user are illustrated in Fig.11.

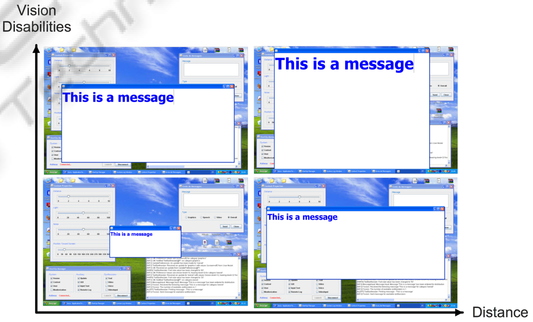

Figure 11: Combined effect on text size of vision capabilities and user's distance to the output device.

Included in the example are four pictures of the system representing different distance values associated with different vision capabilities. It is possible to observe that according to the distance of the user to the output device, the font size dynamically changes. Similarly, it is also possible to observe that the visual capabilities also influence the font size calculation which together with the distance determines the font size.

7 DISCUSSION

Focus on the communication between the system and the user – multimodal output – is the main novelty of our architecture and prototype. In general, systems in this area of application incorporate various modalities to communicate with the user (such as voice, text or images) but they are completely devoid of any autonomy, i.e. simply output the messages they receive using default definitions. In the proposed model, `AdaptO`, it is intended that these modalities have some independence and self-adaptability to user and context of use (ex: environment).

Another distinctive aspect of our proposal is the delegation to the autonomous output agents of the responsibility for registering and, thereafter, of receiving the relevant user and context dynamic attributes necessary for maximizing its adaptability to its destination users.

This design choice has several advantages: It simplifies the fission coordination service, which is no longer required to know everything about all the available output devices. It also easies the introduction of new output agents, making it a more modular and extensible architecture.

However, there are some parts of the output problem that we do not yet address. First, nothing has been done regarding the semantic fission. Second, our user model service is yet a very simplified proof-of-concept engine, composed only of a few user related attributes. Future versions will require not only a strong database infrastructure but also the ability to gather and to learn user related information, such as to register its expressed preferences and to learn from its historical usage patterns. Such historic information is to be systematically registered by an history agent service which is also missing from our current implementation.

A separation between styling and rendering is also missing in out output agents making them less adaptable to different types of output devices (like when we move from a large LCD device to a small smartphone one).

In short, in the WWHT model we have essentially addressed the middle WT parts ("Which" and "How").

8 CONCLUSIONS

The basis for an intelligent adaptation of output - that we called AdaptO, a Portuguese word meaning (I) adapt - was proposed and discussed. First versions of the required services – context and user services – and of a few output agents for text and speech synthesizing are already implemented and are being used for supporting a prototype remote telerehabilitation application.

Ongoing and future work includes: first tests with elderly users (the target for our work); refining adaptation heuristics; use of more advanced user models, possibly making use of ontologies such as GUMO (Heckmann et al., 2005); and creating new output agents such as 3D dynamic graphics and avatars.

ACKNOWLEDGEMENTS

This work is part of the Living Usability Lab for Next Generation Networks (www.livinglab.pt) project, a QREN project, co-funded by COMPETE and FEDER.

REFERENCES

Coetzee, L., Viviers, I., and Barnard, E. (2009). Model based estimation for multi-modal user interface component selection. In *20th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA 2009)*, pages 1–6.

Dumas, B., Ingold, R., and Lalanne, D. (2009a). Benchmarking fusion engines of multimodal interactive systems. In *ICMI-MLMI '09: Proceedings of the 2009 international conference on Multimodal interfaces*, pages 169–176, New York, NY, USA. ACM.

Dumas, B., Lalanne, D., Guinard, D., Ingold, R., and Koenig, R. (2008). Strengths and weaknesses of software architectures for the rapid creation of tangible and multimodal interfaces. In *Proceedings of 2nd international conference on Tangible and Embedded Interaction (TEI 2008)*, pages 47–54.

Dumas, B., Lalanne, D., and Oviatt, S. (2009b). Multimodal interfaces: A survey of principles, models and frameworks. In Lalanne, D. and Kohlas, J., editors, *Human Machine Interaction*, volume 5440 of *Lecture Notes in Computer Science*, pages 3–26. Springer Berlin / Heidelberg.

FIPA (2010 (accessed 7 November 2010)). The foundation for intelligent physical agents. Available: http://www.fipa.org.

Gordon-Salant, S. (2005). Hearing loss and aging: new research findings and clinical implications. *Journal of Rehabilitation Research and Development*, 42(4 Suppl 2):9–24.

Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., and von Wilamowitz-Moellendorff, M. (2005). Gumo - The General User Model Ontology. *User Modeling 2005*, pages 428–432.

JADE (2010 (accessed 7 November 2010)). Jade - java agent development framework. Available: http://jade.tilab.com.

Karpov, A., Carbini, S., Ronzhin, A., and Viallet, J. E. (2008). Two SIMILAR Different Speech and Gestures Multimodal Interfaces. In Tzovaras, D., editor, *Multimodal User Interfaces*, Signals and Communication Technology, chapter 7, pages 155–184. Springer Berlin Heidelberg, Berlin, Heidelberg.

Microsoft (2010 (accessed 18 October 2010)). Developing speech applications. Available: http://www.microsoft.com/speech/developers.aspx.

Rousseau, C., Bellik, Y., and Vernier, F. (2005a). Multimodal output specification / simulation platform. In *Proceedings of the 7th international conference on Multimodal interfaces*, ICMI '05, pages 84–91, New York, NY, USA. ACM.

Rousseau, C., Bellik, Y., and Vernier, F. (2005b). WWHT: un modéle conceptuel pour la présentation multimodale d'information. In *Proceedings of the 17th international conference on Francophone sur l'Interaction Homme-Machine*, IHM 2005, pages 59–66, New York, NY, USA. ACM.

Rousseau, C., Bellik, Y., Vernier, F., and Bazalgette, D. (2004). Architecture framework for output multimodal systems design. In *In Proceeding of OZCHI*.

Rousseau, C., Bellik, Y., Vernier, F., and Bazalgette, D. (2005c). Multimodal output simulation platform for real-time military systems. In *Proceedings of Human Computer Interaction International (HCI International'05)*, Las Vegas, USA.

Rousseau, C., Bellik, Y., Vernier, F., and Bazalgette, D. (2006). A framework for the intelligent multimodal presentation of information. *Signal Process.*, 86:3696–3713.

Teixeira, A., Pereira, C., Oliveira e Silva, M., and Alvarelhão, J. (2011). Output matters! adaptable multimodal output for new telerehabilitation services for the elderly. In *1st International Living Usability Lab Workshop on AAL Latest Solutions, Trends and Applications - AAL 2011 (AAL@BIOSTEC 2011)*. Submitted.