Services to Support Use and Development of Speech Input for Multilingual Multimodal Applications for Mobile Scenarios

António Teixeira, Pedro Francisco, Nuno Almeida, Carlos Pereira, Samuel Silva
Department of Electronics, Telecommunications & Informatics / IEETA
University of Aveiro
Aveiro, Portugal
{ajst, goucha, nunoalmeida, cepereira, sss}@ua.pt

***Abstract*—Speech is our most natural form of interaction. Developing speech input modalities for several languages, combining speech recognition and understanding, presents various difficulties. While automatic translators ease the translation of normal text, the adaptation of grammars for several languages is currently performed based on an *ad hoc* approach. In this paper, we present a novel service that enables a multilingual speech input modality and helps developers in the creation of the grammars for different languages. The service itself uses two additional services for parsing and translation. The use of the service is exemplified in the context of AAL PaeLife project multilingual personal assistant.**

Keywords - service; speech; grammars; multilingual; multimodal interaction; translation.

I. INTRODUCTION

Advances in technology have brought mobile devices to our everyday life. With the growing number of features provided by devices such as smartphones or tablets, it is of paramount importance to devise natural ways of interacting with them that help to deal with their increasing complexity. Natural interaction is, therefore, an important goal, striving to integrate devices with our daily life by using gestures, context awareness or speech.

The importance of natural interaction is also boosted by the needs of various user groups, such as the elderly, that might present some kind of limitation at physical (e.g., limited dexterity) or cognitive (e.g., memory) level and lack the technological skills to deal with devices that can play an important role in improving their daily life [1].

The increased mobility and the multitude of devices that can be used impose important challenges to interaction design. Nevertheless, the "always connected" nature of most of these devices, in a multitude of environments (e.g., home, work and street), offers the possibility of using resources located remotely, including computational power, storage or on-the-fly updates to currently running applications to serve a new context.

Speech and natural language remain our most natural form of interaction [2][3] and a number of recent applications use speech as part of a multimodal system [4] in combination with other modalities. Nevertheless, despite its potential, the inclusion of input and output modalities based on speech poses problems at different levels. On a higher level, speech modalities involve several complex modules that need to work together and ensure speech recognition and speech synthesis. Tailoring these modules to different applications is a tiresome task and we have recently proposed a generic, service-based, modality component [5] that can work decoupled from the application, thus providing easier deployment of speech modalities. Another important issue concerning speech is its inclusion in applications targeting multiple languages. Therefore, our generic modality component also aims at being able to internally handle several languages.

Several well-known applications use speech. A representative example is mTalk [6] multimodal browser developed by AT&T, a tool to support the development of multimodal interfaces for mobile applications. The mTalk uses cloud-based services to process most of the multimodal data. Siri [7] and Google Voice Search [8] are other examples of speech enabled applications, that use cloud based services to process multimodal data.

Automatic Speech Recognition (ASR) takes as input the speech signal and produces a sequence of words. Speech recognition engines are typically based in Hidden Markov Models [9], which provide a statistical model to represent the acoustic model for the utterances. In addition to the acoustic model, a language model or a grammar is also needed to define the language. Language models, such as the ones defined by the ARPA format, are statistical n-gram [10] models that describe the probability of word appearance based on its history. Grammars can be defined as a set of rules and word patterns which provide the speech recognition engine with the sentences that are expected. The Java Speech Grammar Format (JSGF) [11] and GRXML [12] are examples of grammar formats.

Although grammars are more limited in the amount of sentences that will be recognized, they are capable of being more specific to each particular context of use, which often translates to a more accurate recognition.

These models and grammar are language dependent and, therefore, require language specific training. Usually, acoustic models and language models are trained generically to support a broad part of the language. They only need to be trained once for each language. Since grammars are created based on the context of one application, it is necessary to translate the grammars of each application to each language that the application aims to support.

The PaeLife project [13] is aimed at keeping the European elderly active and socially integrated. The project is developing AALFred, a multimodal personal life assistant (PLA), offering the elderly a wide set of services from

 ISBN: 978-1-61208-361-2

unified messaging (e.g., email and twitter) to relevant feeds (e.g., the latest news and weather information). The platform of the PLA comprises a personal computer connected to a TV-like big screen and a portable device, a tablet. One of the key modalities of the PLA is speech; speech input and output will be available in four European languages: French, Hungarian, Polish, and Portuguese. One of the demanding tasks on using the speech modality, due to the several languages involved, is to help developers and user interaction designers in the derivation of the grammars for each language.

Therefore, in this context of multi-language support, our main goals for the generic speech modality include:

- Streamlining of internationalization support;
- Reduce variance among grammars contributing for easier update and maintenance;
- Customization of any of the different grammars, if needed;
- Additionally to manual editing, allow automatic expansion of the recognized sentences and word corpora using existing services.

To approach these goals and in the context of a multimodal personal assistant, AALFred [14], part of the aforementioned project PaeLife, we present a first instantiation of a service which explores automatic translation to provide initial versions for the grammars in the different languages based on the definition of the semantic grammar (in English). The service receives a grammar, translates it and supports the needs of the speech modality.

The multimodal architecture integrating the multilingual support for speech input is directly related to the recent work of the W3C on a distributed architecture for multimodal interaction [15]. In fact, as described in [4][16] we have been working on the application of such architecture to mobile and AAL applications.

The use of services to support the functionalities in speech input has been adopted in several mobile architectures, such as the mentioned mTalk [6] and SIRI [7], but none, to the best of our knowledge, explored the use of automatic translation of grammars to support multilingual speech input.

The remainder of this document is organized as follows: Section II describes the main aspects of the proposed service regarding its architecture and main components; Section III discusses prototype implementation; Section IV provides some application examples; finally, Section V presents some conclusions and ideas for future work.

II. SYSTEM OVERVIEW

The system's main objective is to be able to automatically generate a derived grammar in other target languages. That is achieved by preserving as much of the main grammar structure as possible, generating coherent phrases in the target language and having in consideration the process of word reordering.

The system is dual in functionality. It supports both development and use in real interaction contexts.

In the development stage, developers use the system to make semantic grammars available, to produce the translated versions of such grammars. At this stage the service can also be used remotely to check and make corrections to the grammars. This can be done by native speakers or, if available, language specialists.

In interaction contexts, the system is in charge of the natural language understanding, making use of the grammars sent to the service at development stage. It receives the output of speech recognition and returns the semantic information extracted. The service also returns, on request, to the speech modality, the necessary information on words and sentences needed to configure the speech recognition engine.

A. Architectural Definitions

The architecture, in Fig. 1, is composed of four main components: the speech modality, the core service, the access APIs and the external resources (both parser and translator services). Further details about each component are provided in what follows.

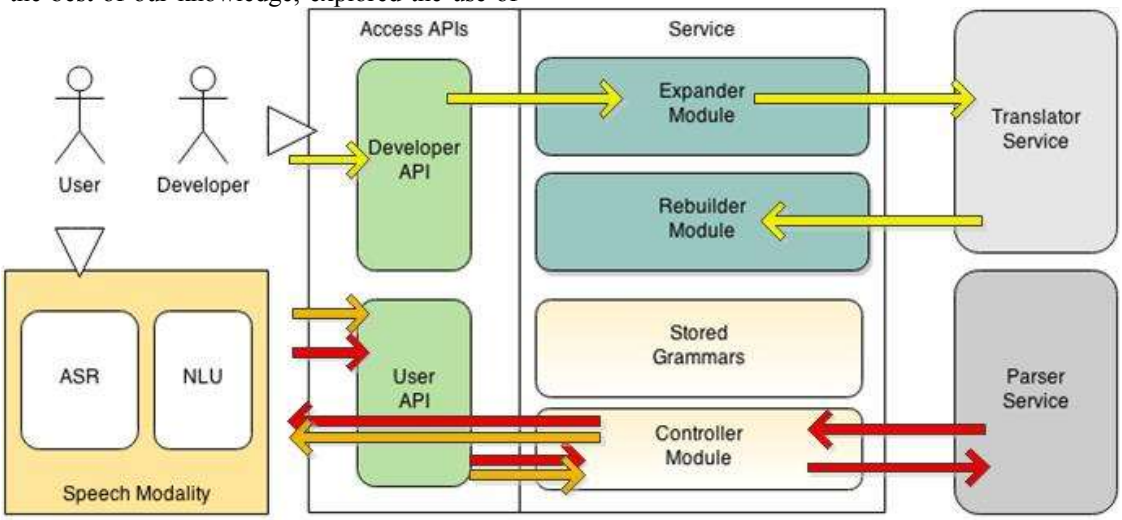

Figure 1 - Conceptual Architecture.

1) Speech Modality

The speech modality allows the recognition of speech in a specified language, previously selected. In practice, speech is processed by the Speech Recognizer (ASR) to produce a list of words that are sent to the Natural Language Understanding (NLU) interpreter to process. The NLU goal is to extract semantic information from the sequence of recognized words. The implementation is aligned with the modalities in a multimodal architecture, integrating in general a recognizer and an interpreter.

2) Main Service

The main service is responsible for the manipulation of the grammar. It allows to: a) upload files and input to be analyzed, and retrieval of the parsing result; b) get all statements generated by the specified grammars and on-demand translation of grammars; c) submit corrections to derived grammars and get a listing of all available grammars.

The service also requires the definition of a format for representation of the input grammar. From this representation, using the Expander Module we are able to generate all possible statements recognizable by the grammar, which are the statements submitted for translation, using the Translator Module.

The service has several ways of being used. The simplest, illustrated in Fig. 2, consists in the submission of a grammar and the selection of an intended language which results in the subsequent generation of valid phrases, to ease the configuration of an ASR by a third party.

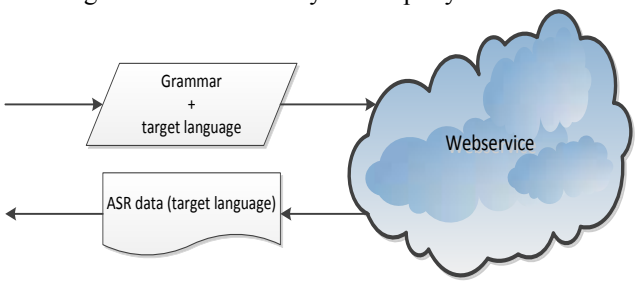

Figure 2 - Simple use of the service to get list of sentences for ASR in a target language.

Figure 3 shows a case where, assuming previous configurations and a working ASR, the service is used to extract semantic tags of a given text, and return them to the caller. This way of using the service implements the multilingual NLU processing.

Given the limitations of automatic translations, the service also supports manual revision and subsequent update of grammars (Fig. 4). This use is particularly suited when developing an application – such as AALFred – allowing the creation of an initial semantic grammar in English and using the service to provide translated grammars in other languages, enabling each involved partner in the project to revise and correct the automatically generated grammars. Each revised version becomes part of the service, after upload, and is used as described in the previous use cases.

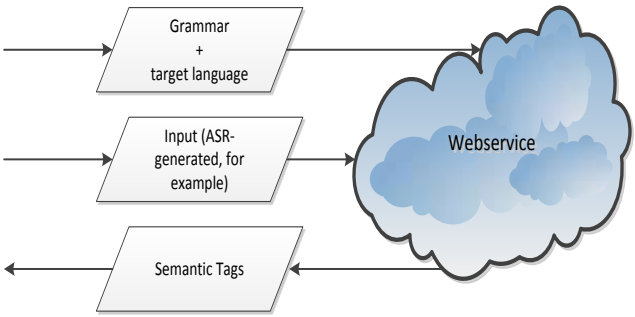

Figure 3 - Service used as multilingual NLU.

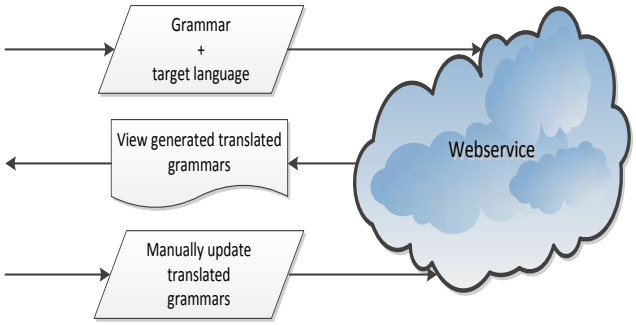

Figure 4 - Service used to manual revision and update of grammars.

After the translation is accomplished, a Rebuilder Module recreates new grammars according to the translated languages. Afterwards, these new grammars are stored within the Stored Grammars module for further usage.

3) Access APIs

All operations are made through the access APIs, ensuring a consistent and complete operation control.

To enable the introduction of new grammars, a specific interface is required for the developer. This interface can be seen as a frontend which allows the developer to submit a grammar and check the results of grammar translation, both in terms of generated grammar and of generated sentences. In our current implementation, it supports editing a grammar and its resubmission. This method enables faster feedback cycles of grammar enhancement.

For the speech modality, a user API is provided, allowing sequences of words from the speech recognizer to be processed in order to obtain semantic tags (i.e., to perform NLU in speech recognizer output).

4) Parser and translator services

The service is connected to two external services. The first one provides parsing; the second provides translation.

 ISBN: 978-1-61208-361-2

III. PROTOTYPE IMPLEMENTATION

To test our architecture and associated ideas, a prototype service has been created and used. Phoenix [17] was chosen as both the parser and grammar specification format. The advantages of this choice are explained by Phoenix's robustness to errors in recognition and parsing abilities. For translation, the choice fell on Bing due to its ability of providing reordering information. Later on, a more detailed explanation will be given on this.

The following sections provide information on the implementation and features of key components within the prototype.

A. Parser service

The objective of the parser is to extract the semantic tags, as defined in the semantic grammar, from the list of words received from the ASR, and return the text plus the semantic tags to be processed by the Interaction Manager and ultimately used by the application.

Internally, the analysis is done by Phoenix. Phoenix uses an automatic translated semantic grammar that allows tags existing on the original grammar to be preserved on the target language grammar.

In order to have an integrated support for the multiple languages of the project – or even other languages – the NLU parser is coupled with the management and process of automatic derivation of grammars by automatic translation.

B. Translation of Semantic Grammars

The goal is to translate to a target language all the terminal words while preserving the semantic tags. Translation must also produce a complete list of sentences defined by the grammar.

The process adopted and implemented is composed of three steps: 1) full expansion of the grammar; 2) translation; and 3) grammar rebuild.

1) Grammar Expansion

In order to be able to manipulate the Phoenix Grammar, one of two approaches had to be followed: either change the Phoenix Parser or have a separate parser to parse the Phoenix Grammar structures onto a separate data structure, on which we would then apply our modifications. We decided to implement a separate parser so as not to change the Phoenix code, allowing us to use C# for our work and rely on the Phoenix Parser only for its already defined and well-tested function: parsing the input text based on a defined Grammar.

In order to properly translate the grammar to take in consideration word reordering, we need to submit the full sentence for the translator to properly evaluate which translation to provide. While a word-by-word translation would yield a non-natural result, submitting the whole sentence allows us to retrieve a translated sentence that sounds natural and takes in consideration language specific connectors and variances which may not exist on the original language.

The algorithm developed makes use of two data structures: an "in progress" stack and a "done so far" queue. On the first, the algorithm stores the current rule while on the second it stores the translated words. Expanding all the rules is done by keeping the history of the rules visited along the expansion.

2) Translation

The translation process consists in submitting the result of the expansion (words plus their history/grammars rules) and receiving the resulting translated sentences (pairing of words in the translation with the correspondent words in the source).

In our prototype, we selected Bing Translator as the translator service. The usage of the Bing Translator is an advantage to us since it provides the realignment info [18] necessary to get word reordering support during the grammar rebuild process. That realignment info both eases the matching of translation with source words and is what allows us to support word reordering when reconstructing grammar rules. In addition, Bing Translator also allows us to obtain multiple translations per request, which enables the expansion of an existing grammar to support several similar sentences, with no need of additional input by the developer. We can thus increase the coverage of our grammar in an automatic and effortless way.

TABLE 1 – EXAMPLE OF BING TRANSLATION REORDERING INFO.

Source text: The answer lies in machine translation. Translated Text: La réponse se trouve dans la traduction automatique. Alignment info: 0:2-0:1 4:9-3:9 11:14-11:19 16:17-21:24 19:25-40:50 27:37-29:38 38:38-51:51 The -> La answer -> réponse lies -> se trouve in -> dans machine -> automatique translation -> traduction . -> .

3) Grammar Rebuild

When the grammar is parsed (in order to expand it afterwards), a different object is created for each instance of any rule. As such, for each Terminal Word present in the statement resulting from the expansion of the grammar, we can determine exactly which rule gave origin to the path that lead to it after the sentence is submitted for translation. Since we have reordering info available, we know which rules generated the text resulting from the translator.

The developed algorithm uses the saved Grammar Expansion history and the translated sentences of the Translation Process. It consists of analyzing the ancestors' historic information to remake the grammar. This is done by merging Non-Terminals of the same level throughout the

 ISBN: 978-1-61208-361-2

grammar in a top-bottom approach. Fig. 5 and 6 show an example.

[Main]	[Main]	[Main]	[Main]
[OPEN_NEWS]	[OPEN_NEWS]	[OPEN_NEWS]	[OPEN_NEWS]
A	[ITEM_NUM]	Elem	Megnyitása
	Második		

Figure 5 - Initial representation of the grammar.

[Main]			
[OPEN_NEWS]			
A	[ITEM_NUM]	Elem	Megnyitása
	Második		

Figure 6 - Resulting data after application of rebuild algorithm.

Duplicates are eliminated automatically, thus obtaining the grammar according to the translation given.

IV. First Results

Currently, the developed service module supports the translation of text in English to French, Hungarian, Polish and Portuguese. Furthermore, it supports translations from French, Hungarian, Polish and Portuguese to English. Two examples of service usage are presented in this section.

A. Example of grammar translation

After the submission of a new grammar, either via a direct API or via a website (in development), the submitted grammar will be parsed and stored in memory after which all phrases will be generated. As an example, the grammar in Fig. 7 will be converted to the Hungarian translation presented in Fig. 10.

```
[Main]
        ([VIEW_DAYS])
        ([OPEN_NEWS])
;
[DAY]
        (yesterday)
        (today)
;
[ITEM_NUM]
        (first)
        (second)
        (third)
;
[VIEW_DAYS]
        (news from [DAY])
        (open [DAY] news)
;
[OPEN_NEWS]
        (open the [ITEM_NUM] item)
;
```

Figure 7 – Example of original grammar (in English) sent to the service by an application developer.

```
news from yesterday
news from today
open yesterday news
open today news
open the first item
open the second item
open the third item
```

Figure 8 - Result from the expansion of the original grammar (in English).

```
a tegnapi hírek
a mai hírek
nyissa meg tegnap hírek
nyissa meg mai hírek
nyissa meg az elso elemet
a második elem megnyitása
a harmadik elem megnyitása
```

Figure 9 - Results from translation of the sentences in Fig.8 to Hungarian.

```
[Main]
        ([VIEW_DAYS])
        ([OPEN_NEWS])
;
[DAY]
        (tegnapi)
        (mai)
        (tegnap)
;
[ITEM_NUM]
        (elso)
        (második)
        (harmadik)
;
[VIEW_DAYS]
        (a [DAY] hírek)
        (nyissa meg [DAY] hírek)
;
[OPEN_NEWS]
        (nyissa meg az [ITEM_NUM] elemet)
        (a [ITEM_NUM] elem megnyitása)
;
```

Figure 10 - The resulting Hungarian grammar.

As can be seen following the steps, the grammar in English is used to generate all sentences (Fig. 8), which are then translated. The translation (Fig. 9) is then used, in conjunction with word generation history, to rebuild the grammar in Hungarian, with flexibility to deal with word reordering (in bold) and to synonyms/alternatives (underlined).

B. An example of grammar manual fine tuning

The system autonomously generates a grammar ready to be used on any language. However, it is possible to fine-tune the grammar to achieve a higher degree of correctness. This can be done by the developer or by a third party. The web based grammar editor allows previewing the sentences that the edited grammar describes and resubmission of the grammar (Fig. 11). To complement the first example in Hungarian, this example is in French.

 ISBN: 978-1-61208-361-2

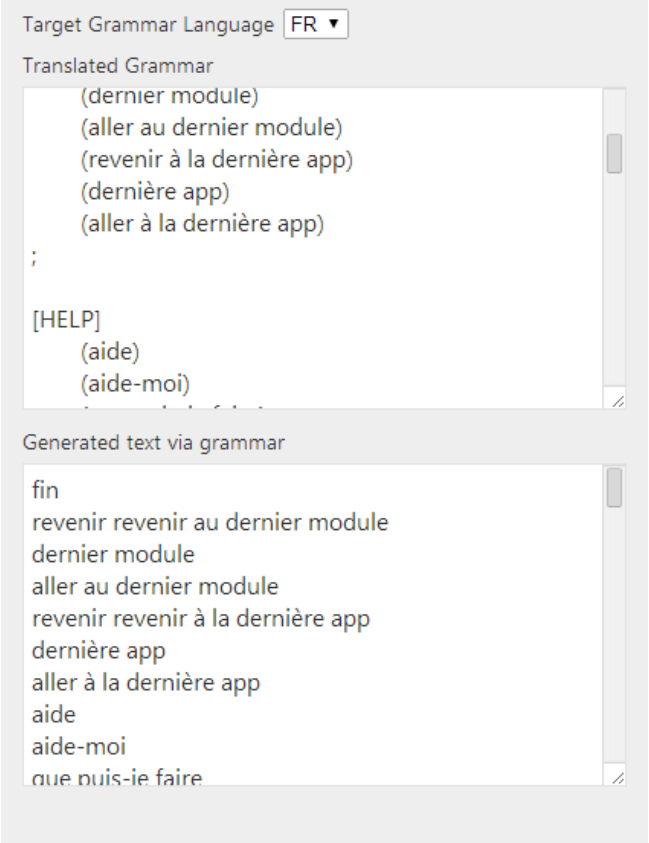

Figure 11 - Web base editor showing the translation to French in manual edition and the corresponding list of sentences defined by the edited grammar. Some generation problems are noticeable, such as repetition of word "revenir".

V. CONCLUSIONS AND FUTURE WORK

Multilanguage support in speech modalities is a complex task. In the context of a generic service-based speech modality, proposed by the authors, a service is presented which aims to provide support for easy deployment of applications supporting several languages. The main highlight of the proposed service is the possibility to generate grammars for different languages by automatic translation of an existing grammar (in English). A first prototype has been implemented and tested and several application examples are provided.

Future developments should explore the use of multiple translation services, increasing the probability of having, in the set of translated sentences, the correct ones. The evaluation in real use, both by users of the personal assistant integrating the speech modality and developers, will be performed in the next months, as part the development process in project PaeLife.

ACKNOWLEDGMENT

Authors acknowledge the funding from AAL JP, FEDER, COMPETE and FCT, in the context project of AAL/0015/2009, project AAL4ALL (www.aal4all.org), IEETA Research Unit funding FCOMP-01-0124-FEDER-022682 (FCT-PEstC/EEI/UI0127/2011) and project Cloud Thinking (funded by the QREN Mais Centro program, ref. CENTRO-07-ST24-FEDER-002031).

REFERENCES

[1] T. H. Bui, "Multimodal Dialogue Management - State of the art" , Technical Report no. TR-CTIT-06–01, 2006.

[2] N. O. Bernsen, "Towards a tool for predicting speech functionality," Speech Communication, vol. 23, no. 3, pp. 181–210, 1997.

[3] C. Munteanu et al., "We need to talk: HCI and the delicate topic of spoken language interaction," in CHI'13 Extended Abstracts on Human Factors in Computing Systems, April 2013, pp. 2459–2464.

[4] A. Teixeira et al., "Multimodality and Adaptation for an Enhanced Mobile Medication Assistant for the Elderly," in Proc. Third Mobile Accessibility Workshop (MOBACC), CHI 2013, April 2013.

[5] N. Almeida, S. Silva and A. Teixeira, "Design and Development of Speech Interaction: A Methodology", in Proc. HCI International, 2014, in press.

[6] M. Johnston, G. Di Fabbrizio and S. Urbanek, "mTalk - A Multimodal Browser for Mobile Services.," in Interspeech, 2011, pp. 3261–3264.

[7] "iOS - Siri.", http://www.apple.com/ios/siri/ [Accessed: 21-Mar-2014].

[8] "Voice Search", http://www.google.com/insidesearch/features/voicesearch/index-chrome.html [Accessed: 21-Mar-2014].

[9] B. Singh, N. Kapur and P. Kaur, "Speech Recognition with Hidden Markov Model: A Review," Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 2, no. 3, pp. 400–403, 2012.

[10] P. Clarkson and R. Rosenfeld, "Statistical language modeling using the CMU-cambridge toolkit", Eurospeech, 1997, vol. 97, pp. 2707–2710.

[11] T. Brøndsted, "Evaluation of recent speech grammar standardization efforts.," in Interspeech, 2001, pp. 1089–1092.

[12] A. Hunt and S. McGlashan, "Speech Recognition Grammar Specification Version 1.0", http://www.w3.org/TR/speech-grammar/ [Accessed: 21-Mar-2014].

[13] "PaeLife: Personal Assistant to Enhance the Social Life of Seniors.", http://www.microsoft.com/portugal/mldc/paelife/ [Accessed: 21-Mar-2014].

[14] A. Teixeira et al., "Speech-Centric Multimodal Interaction for Easy-To-Access Online Services : A Personal Life Assistant for the Elderly," Procedia Computer Science, pp. 389–397, 2013.

[15] D. A. Dahl, "The W3C multimodal architecture and interfaces standard," J. Multimodal User Interfaces, vol. 7, no. 3, pp. 171–182, Apr. 2013.

[16] A. Teixeira, N. Almeida, C. Pereira and M. Oliveira e Silva, "W3C MMI Architecture as a Basis for Enhanced Interaction for Ambient Assisted Living," in Get Smart: Smart Homes, Cars, Devices and the Web, W3C Workshop on Rich Multimodal Application Development [online], July 2013.

[17] W. Ward, "Understanding spontaneous speech: the Phoenix system," in International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1991, pp. 365–367.

[18] "Word Alignment Information from the API", http://msdn.microsoft.com/en-us/library/dn198370.aspx [Accessed: 21-Mar-2014].

 ISBN: 978-1-61208-361-2