Services to Support Use and Development of Multilingual Speech Input

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

Abstract—**The use of speech for human-machine interaction benefits from being our most natural way of communication. Profiting from a widespread use of devices providing audio input, applications increasingly support speech recognition and understanding, but several challenges are still to be addressed. A common solution to support spoken language understanding uses semantic grammars. Also, for speech recognizers, grammars are a common way of configuring what can be recognized. The definition of these grammars, to support a semantically rich spoken interaction, involves a large amount of resources and linguistic knowledge and, therefore, systems tend to support a single language or multiple languages in a very narrow semantic scope. To address this issue, we propose to use an existing grammar to generate grammars for other languages by using automatic translation and taking into consideration aspects such as word realignment and grammar expansion, based on multiple possible translations. This method fosters effortless creation of a semantically rich grammar, for the target language, which can then be revised. The proposed method has already been successfully used to enable multilingual speech interaction for AALFred, a personal life assistant, covering English, French, Hungarian, Polish, and Portuguese.**

Keywords - Services; speech; semantic grammars; multilingual; multimodal interaction; speech recognition, 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 [1].

The importance of natural interaction is also boosted by the needs of various user groups, such as the elderly, who 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 [2].

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 wide range 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 [3][4] and a number of recent applications use speech as part of a multimodal system [5] 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 [6] 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. Since configuration information for speech recognizers or extraction of semantic information is created based on the context of one application, it is necessary to explicitly define the grammars of each application in each language that the application aims to support.

One of the demanding tasks on using the speech modality, when several languages are 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;
- Reduction of 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 [7], part of project PaeLife, we proposed [1] the use of an existing grammar to generate grammars for other languages by using automatic translation and developed a first instantiation of a service which explores this idea 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 option for a service based solution follows a strong recent trend. Many mobile applications adopt cloud services, extending the capabilities of the device [8] regarding storage and processing capabilities. Also, the use of services to support the functionalities in speech input has been adopted in several mobile architectures, such as the mentioned mTalk [9], SIRI [10], and Cortana [11]. 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 presents some background information regarding the application scenario in speech technologies, multimodal architectures, and support for spoken interaction in such architectures; Section III describes the main aspects of the proposed service regarding its architecture and main components; Section IV discusses prototype implementation; Section V provides some application examples; finally, Section VI presents some conclusions and ideas for future work.

II. BACKGROUND AND RELATED WORK

To help understand the proposed service and how it relates to recent work in the areas of multimodal interaction and spoken language interaction, some background information and related work are presented. Also, to start this section, we briefly present the applications scenario chosen for the proof-of-concept is made.

A. The application Scenario: The AAL PaeLife project

The PaeLife project [12], chosen as the application scenario, is aimed at keeping the European elderly active and socially integrated. The project developed AALFred [7], [13], a multimodal personal life assistant (PLA), offering the elderly a wide set of services from 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 interaction modalities of the PLA is speech; speech input and output is available in five European languages: French, Hungarian, Polish, English, and Portuguese.

B. Speech and interaction

Speech and natural language remain our most natural form of interaction [3][4] and a number of recent applications use speech as part of a multimodal system [5] 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 [6] 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 [9], a multimodal browser developed by AT&T, to support the development of multimodal interfaces for mobile applications. Siri [10] and Google Voice Search [14] are other examples of speech enabled applications.

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 [15], 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 [16] 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) [17] and GRXML [18] 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.

The next phase after speech recognition is Spoken Language Understanding (SLU), having as goal to extract semantic information from the sequence of words produced by the speech recognizer. Even in a command and control scenario for speech interaction it is very useful to associate a semantic meaning such as “direction left” to the sequence of words “please turn to the left”. Several different types of SLU [19][20] have been proposed, which can be divided into major groups: knowledge-based and data-driven approaches. Knowledge-based solutions include semantically enhanced syntactic grammars, and semantic grammars, whereas data-driven approaches explore both generative models and conditional (nongenerative) models [19]. Despite the potential of data-driven approaches, it is common to use semantic grammars (a knowledge-based approach) in many applications. These approaches represent the semantic space by a set of templates represented by semantic frames. Each frame contains typed components called “slots” (or frame elements). The type of the slot specifies what kind of fillers it expects. The goal of this frame-based SLU is to choose the frame that best matches the sequence of words coming from the speech recognizer and extract the values for the slots [19].

C. The Multimodal Architecture and Speech Modality

The support for multilingual spoken language interaction must be part of the multimodal architecture supporting interaction by voice. We have been working on the

application of such architecture - directly related to the recent work of the W3C on a distributed architecture for multimodal interaction [21] - to mobile and AAL applications as described in [5][22].

One of the major advantages of this architecture is the decoupled nature of the interaction modalities, which enables the development of the application core without an explicit consideration of which modalities might be involved. What strictly matters is the semantic content resulting from the interactions, i.e., it does not matter if the user says "Go left" or presses the left arrow key in a keyboard. The application receives 'LEFT' for both. In this context, in a multilingual scenario, the application core stays the same and the support for new languages is added to the speech modality, which makes it easier to improve, in parallel with application development, or at a later stage, profiting from more advanced processing methods. The decoupled nature of this generic speech modality also enables its use in any application adopting the multimodal architecture.

III. 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 and 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 SLU, 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.

1) Speech Modality

The speech modality is aligned with modalities in the multimodal architecture. It is a generic modality [6] able to provide speech interaction to applications that adopt the Multimodal Framework [23].

The modality allows the recognition of speech in a specified language, which can be changed at any time[1].

When it starts or the language is changed, the modality

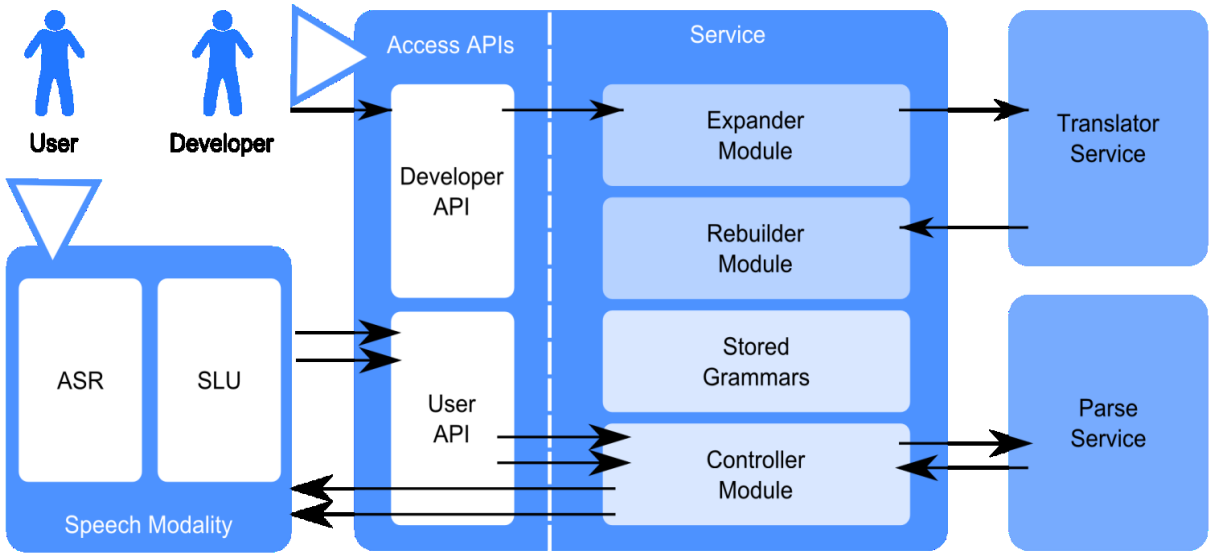

Figure 1. Overall architecture of the presented grammar generation service. The speech modality communicates with the core service, through specific APIs and external services are used for translation and parsing.

[1] It is mandatory that the speech engine and the support for the selected language are installed.

requests the service for the corresponding grammar, identified by an id and the desired language. The service answers with a GRXML grammar, which is directly loaded to a speech recognizer (ASR) engine and then waits for the user speech input. Grammars can be updated by a dynamic rule, which is the task addressed by the service described in the paper.

Each time the user speaks and a sentence is recognized by the speech engine, if the parameter of confidence of the recognition is bigger than a configured threshold, the modality requests the service to process that sentence in the SLU interpreter. The SLU extracts the semantic information of the sentence and sends it back to the modality, which relays its contents to the application through the interaction manager. Although grammars in the modality are in GRXML format, the translation service requires Phoenix grammars [24][25].

The Phoenix grammar is divided into frames containing slots and all slots start with a name and end with a semi-colon. The name of the slots is between square brackets. Inside slots any item string is between parentheses. Items can be of three types, the actual words to be recognized, other slots names, and variables. The information between parenthesis define the patterns for filler strings that can be accepted and are converted by Phoenix into recursive transition networks (RTNs) and are equivalent to context free grammars [19]. These patterns can also be seen as rules.

The difference between slots and variables are that variables are defined inside a slot and can only be used inside that slot. Finally, asterisks define an optional word. Fig. 2 is an example of the construction of a phoenix grammar.

```
[SlotName]
   (word1 word2 [other_slot] VARIABLE1)
   (word3 VARIABLE2 *word4 VARIABLE1)

VARIABLE1
   (word5)
   (word6 word7)

VARIABLE2
   (word8)
   (word9)
;
```

Figure 2. Example of a Phoenix grammar, From: From:http://wiki.speech.cs.cmu.edu/olympus/index.php/Phoenix_Grammar Reference

The construction of the grammar will be presented in Section IV, Subsection B – 2).

2) Main Service

The main service is responsible for the manipulation of the grammar. It allows: a) uploading files and input to be analyzed, and retrieving parsing results; b) getting all sentences generated by the specified grammars and on-demand translation of grammars; c) submitting corrections to derived grammars; and d) retrieving a list of all available grammars.

Overall, the service supports three usage scenarios. The simplest, illustrated in Fig. 3, consists in the submission of a grammar and the selection of an intended language, which results in the subsequent generation of a different GRXML grammar, supported by speech modality.

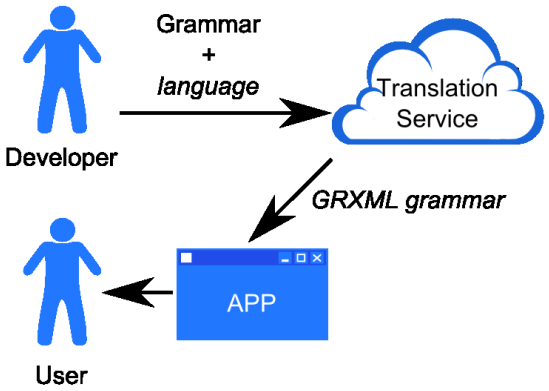

Figure 3. Simple use of the service to get list of sentences for ASR in a target language.

Fig. 4 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 SLU processing.

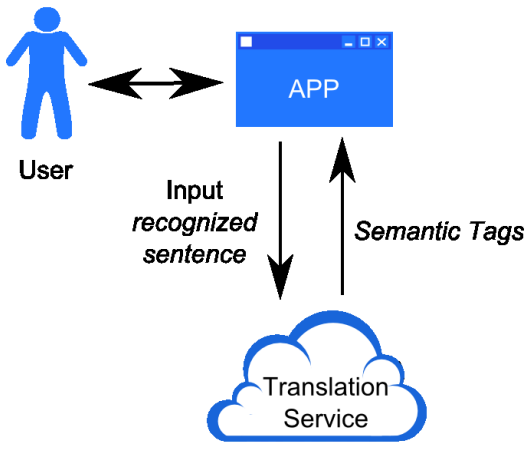

Figure 4. Service used as multilingual SLU.

Given the limitations of automatic translations, the service also supports manual revision and subsequent update of grammars (Fig. 5). 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.

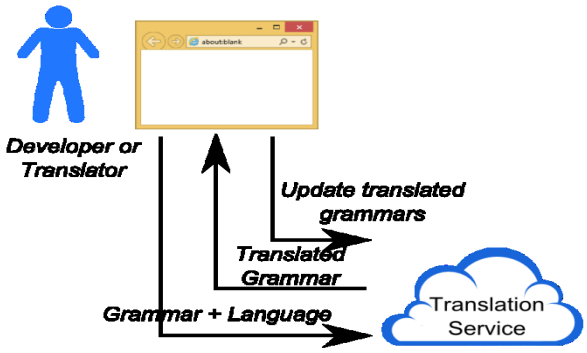

Figure 5. Service used to manual revision and update of grammars.

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 insertion 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 SLU in speech recognizer output).

4) *Parser and translator services*

The service is connected to two external services. The first one provides parsing of the word sequences resulting from the speech recognition process; the second provides translation of sentences.

IV. SERVICE IMPLEMENTATION

To test our architecture and associated ideas, a service has been created and used. Phoenix [24] 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 SLU 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 to the translator. 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.

To enable this sentence based translation we need to obtain all the sentences represented by the Phoenix grammar. This is done by a complete expansion of the grammar, replacing all the non-terminals by all the possibilities.

Without entering into code details, not the aim of this paper, the expansion is made by using a recursive method that makes use of three data structures: a list with all the rules that constitute the grammar, named **remainingList**; an

inProgress stack; and a **doneSoFar** queue. At the start, the grammar is parsed, the list of all rules created and the recursive expansion method invoked with this list as **remainingList** and an empty **doneSoFar** queue. The recursion starts by identifying the MAIN rule of the grammar and processing it.

Processing of a rule consists essential of: (1) adding, to the **remainingList,** the lines including non-terminals; (2) calling again the method to process the next terminal or non-terminal in the rule in processing; (3) if a terminal symbol is detected it is added to the **doneSoFar** queue; (4) if the symbol to be processed is a non-terminal, possible replacement values are added to the rule in processing and a recursive call is made.

During the expansion of all the rules, the history of the rules visited along the expansion is kept, and used in the grammar rebuilding process (step 3, explained ahead).

2) *Translation*

The translation process consists in submitting the result of the expansion and receiving the resulting translated sentences and the information regarding the 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 [26] 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.

The translation of a sentence obtained in the grammar expansion process is illustrated in Fig. 6.

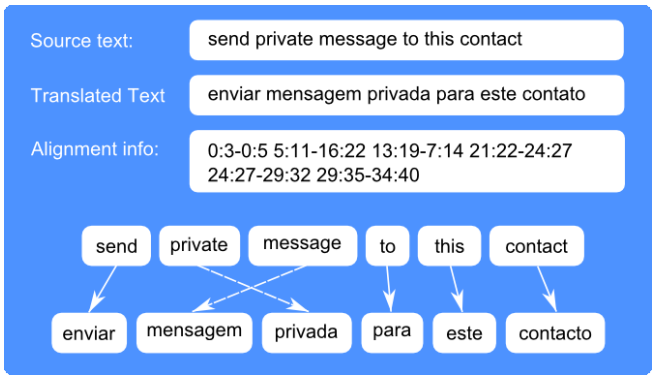

Figure 6. Example of the translation process of a sentence obtained by grammar expansion. In this case, the original sentence (Source Text) is in English and was translated using Bing to Portuguese (Translated Text). Additionally to the translated sentence is provided information on the alignment, which, for easier interpretation, is represented graphically at the bottom of the figure.

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 in analyzing the ancestors' history information to remake the grammar. This is done by merging Non-Terminals of the same level throughout the grammar in a top-bottom approach. Figs. 7 and 8 show an example. Fig. 7 presents, in 2 columns, the input grammar in English and the resulting Portuguese grammar. At the bottom of this figure are presented examples of the sentences resulting from grammar expansion (at left) and the sentences obtained by translation. How the translation results are used to create the translated grammar is illustrated in Fig. 8. After obtaining the representation shown in the figure, duplicates are eliminated automatically, thus obtaining the grammar according to the translation given.

[Main] ([AGENDA]) ; [AGENDA] (*view [WEEKDAYS] *schedule) ; [WEEKDAYS] ([MONDAY]) ([TUESDAY]) ; [MONDAY] (monday) ; [TUESDAY] (tuesday) ;	[Main] ([AGENDA]) ; [AGENDA] (*ver *calendário [WEEKDAYS]) ; [WEEKDAYS] ([TUESDAY]) ([MONDAY]) ; [MONDAY] (segunda-feira) ; [TUESDAY] (terça-feira) ;
monday tuesday monday schedule tuesday schedule view monday view tuesday view monday schedule view tuesday Schedule	segunda-feira terça-feira calendário segunda- feira calendário terça- feira ver segunda-feira ver terça-feira ver calendário segunda-feira ver calendário terça-feira

Figure 7. Example illustrating part of the grammar translation process, showing the input grammar in English, at left, and the resulting Portuguese grammar. At the bottom of the figure are presented examples of the sentences resulting from grammar expansion (at left) and the sentences obtained by translation.

[Main] [AGENDA] Ver	[Main] [AGENDA] Calendário	[Main] [AGENDA] [WEEKDAYS] [MONDAY] segunda-feira

Figure 8. Illustration of the process of grammar reconstruction for a sentence of the example presented in previous figure.

C. *Dynamic Rules*

In some cases, we might also need to have dynamic content in the grammars. For example, if we want to be able to select contacts by their names, we must allow a way to inform the system of the name of a new contact.

These dynamic rules must be created as all the other rules, but the developer does not provide the complete information for these rules. Also, they are not translated. These rules remain empty until the application needs to recognize dynamic content. When this happens, the application requests the service to update the current grammar. To do it, the application sends the identification of the grammar, the rule and the list of sentences to be inserted in that rule. After that, the service processes the new grammar and provides an updated version of the GRXML grammar for the Speech Recognition.

Fig. 9 illustrates the messages between application and service.

D. *Manual Revision of the Translated Grammars*

As automatic translation is not always accurate, the possibility of manually editing the automatic translation results was made possible by creating a website for revision to the grammars (Fig 10 shows a screenshot of the website). The process of revision is very simple. The translator can access a grammar and then choose the language to review. In the review page, there is the possibility to generate all the possible sentences and analyze the correctness of the sentences. Having identified errors in sentences, it is easy to find and correct the grammar. To simplify the process, translated grammar and original appear side by side. In addition to correct sentences, the human editor can also delete or add new possible sentences since they do not delete or change the rule names. At the end, before storing the changes, the grammar is validated to confirm that there are no syntax errors.

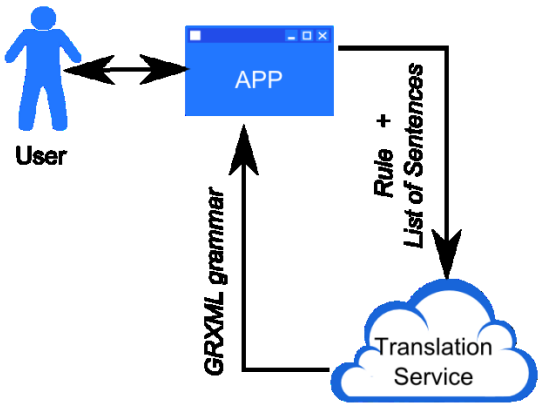

Figure 9. The messages between application and service when using the process of dynamic update of a grammar rule. First, the application sends a list of sentences or words to be added to a specified rule; after this, the service recreates the Grammar for the speech recognizer and sends it to the application.

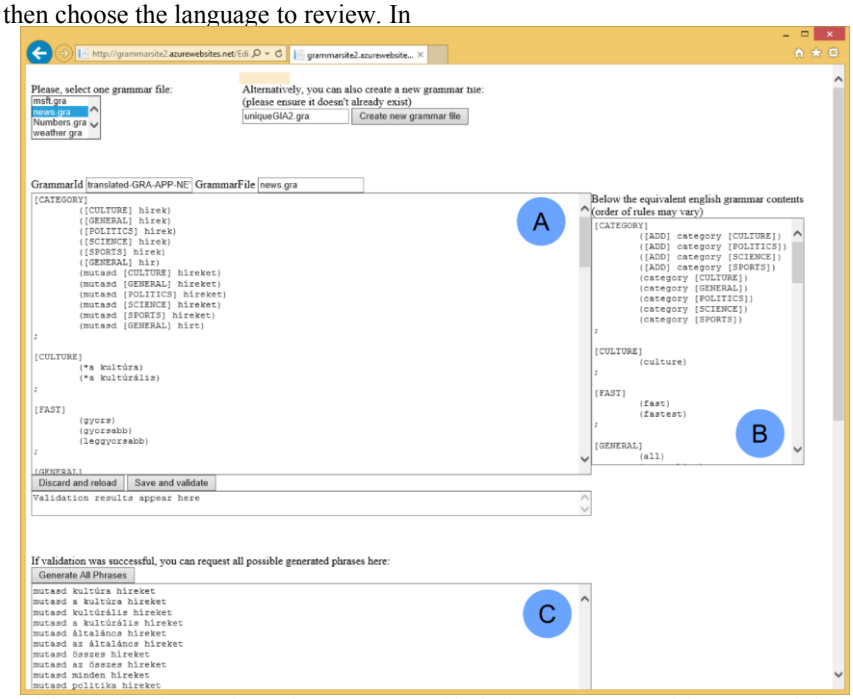

Figure 10. Screenshot of the user interface used to review grammars and including: (A) contents of the grammar being revised; (B) contents of the reference (English) grammar; and (C) a thorough list of all possible phrases generated based on the grammar in (A) for verification purposes.

V. RESULTS

Currently, the developed service module supports the translation of text from English to French, Hungarian, Polish, and Portuguese. Furthermore, it supports translations from French, Hungarian, Polish, and Portuguese to English.

We start this section by presenting some illustrative examples of the service usage. Further on, we present information regarding real results of the adoption of this service for the multimodal multilingual Personal Assistant AALFred.

A. Representative Examples of Use

Illustrating service usage, three examples are presented for: (1) grammar translation; (2) grammar manual revision; and (3) dynamic definition of part of a grammar.

1) Example of grammar translation

After the submission of a new grammar to the service, it will be parsed and stored in memory after which all phrases will be generated. As an example, the grammar in Fig. 11 is converted to the Hungarian translation presented in Fig. 14.

As can be seen following the steps, the grammar in English is used to generate all sentences (Fig. 12), which are then translated. The translation (Fig. 13) 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).

```
[Main]
        ([ACTION])
        ;
[ACTION]
        ([GENERICENTITY])
        ;
[GENERICENTITY]
        ([NAVIGATION])
        ;
[NAVIGATION]
        (go [DIRECTION])
        (scroll [DIRECTION])
;
[DIRECTION]
        ([DOWN])
        ([LEFT])
        ([RIGHT])
        ([UP])
;
[DOWN]
        (down)
;
[LEFT]
        (left)
;
[RIGHT]
        (right)
;
[UP]
        (up)
;
```

Figure 11. Example of original grammar (in English) sent to the service by the developers of the News module of AALFred.

```
go down
go left
go right
go up
scroll down
scroll left
scroll right
scroll up
```

Figure 12. Result from the expansion of the original grammar (in English).

```
menjen balra
menjen felfelé
menjen le
menjen lefelé
menjen jobbra
balra görgetéshez
felfelé görgetéshez
le görgetéshez
lefelé görgetéshez
jobbra görgetéshez
görgetés balra
görgetés felfelé
görgetés le
görgetés lefelé
görgetés jobbra
menj balra
menj felfelé
menj le
menj lefelé
menj jobbra
lapozzunk balra
lapozzunk felfelé
lapozzunk le
….
```

Figure 13. Results from translation of the sentences in Fig.12 to Hungarian.

```
[Main]
        ([ACTION])
        ;
[ACTION]
        ([GENERICENTITY])
        ;
[GENERICENTITY]
        ([NAVIGATION])
        ;
[NAVIGATION]
        (menjen [DIRECTION])
        (**[DIRECTION] görgetéshez**)
        (görgetés [DIRECTION])
        (menj [DIRECTION])
        (lapozzunk [DIRECTION])
        (felmegy)
        ;
[DIRECTION]
        ([LEFT])
        ([UP])
        ([DOWN])
        ([RIGHT])
        ;
[LEFT]
        (balra)
        ;
[UP]
        (felfelé)
        ;
[DOWN]
        (le)
        (lefelé)
        ;
[RIGHT]
        (jobbra)
        ;
```

Figure 14. The resulting Hungarian grammar.

2) 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. Each partner revised the translation of is language. In Fig. 15, the automatic translation and the human revised version of the previous example are shown for French.

```
[NAVIGATION]
        (défilement [DIRECTION])
        (aller [DIRECTION])
        (faites défiler [DIRECTION])
        (défilez [DIRECTION])
        (allez [DIRECTION])
        ;
```

```
[NAVIGATION]
        (défilement [DIRECTION])
        (aller à [DIRECTION])
        (faites défiler à [DIRECTION])
        (défilez à [DIRECTION])
        (allez à [DIRECTION])
        ([DIRECTION])
        (défilement à [DIRECTION])
        (défiler à [DIRECTION])
        (va à [DIRECTION])
        (vers la [DIRECTION])
        ;
```

Figure 15. Example of an automatic translated rule (at top) and its revised version (at bottom)

3) Example of dynamic definition of part of a grammar

In AALFred each user has different contacts and there is the need to recognize the name of the contacts. For that, the "NAMES" rule is updated in runtime. Fig. 16 presents the initial grammar (with only a placeholder) and the updated grammar with the names of contacts sent by AALFred (in this case the authors of this article).

```
[NAMES]
      (zxzxzxzx)
;
```

```
[NAMES]
    (António Teixeira)
    (Pedro Francisco)
    (Nuno Almeida)
    (Carlos Pereira)
    (Samuel Silva)
;
```

Figure 16. Dynamic rule content; (left) initial grammar fragment; (right) updated grammar with names

B. Real Application Example - Supporting AALFred

The service presented was evaluated and continuously evolved in the development and field trials of a real application, the already mentioned Personal Assistant of Paelife project. With the proposed methods we aimed to provide developers with the tools needed to deploy applications supporting speech interaction in multiple languages. In this context, the achievements of project Paelife, involving a multinational consortium and supporting speech interaction in five different languages is, in our opinion, the most illustrative and significant result that can be reported.

1) AALFred modules

The development of AALFred has involved several partners, from different countries and languages, and each partner developed one or more modules. Each module provides users with a number of features enabling access to different content, from social messages, agenda, contacts, places of interest, news, and weather. Each partner created a fragment of a grammar to support the module. Table I shows examples of commands for each module.

TABLE I. MODULES INTEGRATING AALFRED AND EXAMPLES OF COMMANDS ACCEPTED FOR EACH ONE.

AALFred Module	Examples of commands integrating the grammar for the module [English version]
Messages	Open first message Delete selected message Replay to this message Send new email
Agenda	Open Monday Add new appointment Delete selected appointment
Contacts	Change photo Edit this contact Call him
Places of Interest	Find bus station Zoom in Find services near me
News	Category sports Open second item Read content
Weather	Five days forecast Show map Choose location

2) Grammar

In order to organize the grammar and have a structured analysis of the semantic output, we defined a format and structure which any developer needed to follow. By doing this, setting rule names and how the semantic output will be gets clearer. This structure also provides a unification of the semantic output from the various modules, making it easier for the application developer. The convention adopted is based on the Speech Acts [27][28].

For unification purposes, Main must be the first rule. It will consist of several rules that derive directly from Speech Acts, namely:

- **ACTION** – to specify all the actions in the grammar. Inside this rule must be specified all the other rules that

represent entities that can be affected by an action, like Contacts or Messages for example.
An action obligates the listener (in our case AALFred) to either perform the requested action or communicate a refusal or inability to perform the action.

- **INFORM** – used to communicate information to the listener. For example to specify the value of a setting in the app.
- **HELP** – exclusively reserved for the helping system that is being developed for AALFred. It will provide tips to the user based on his expertise on the app.

In most of the cases, an action will be referring to a specific entity. For example, delete can refer to a contact or a message. But, there are also some actions, like "OK" or "CANCEL", which do not refer to any entity in specific. Yet we have to associate them to an entity and that is where GENERICENTITY comes in. Despite its usefulness, we tend to avoid the use of GENERICENTITY as the implementation in the application is harder since it obligates to be aware of the context.

As mentioned, ACTION needs the specification of an entity. Examples of entities in AALFred context are: Contacts, News, and Pharmacies. Each entity has a number of attributes.

To provide the application developers with a simple and unified semantic output we adopted the following organization – with direct implication on the output obtained from the semantic parser - was adopted:

Main
→ Acts (Action | Inform | Help)
→ Entity
→ Parameters

Example of usage of the proposed convention: Considering the excerpt of grammar presented in Fig. 17, related to a news module. It allows the selection of the categories of the news. In this case, if the user says "category sports", the output of the semantic analysis is:

Main → ACTION → NEWS → CATEGORY → SPORTS

Meaning that an Action must be performed and that the entity is the News and the parameter is "Sports". Also, every action related to the news modules (an Entity) starts with Main-ACTION-NEWS.

```
[Main]
        ([ACTION])
        ([INFORM])
        ([HELP])
;
[ACTION]
        ([GENERICENTITY])
        ([NEWS])
        …
;
[NEWS]
        ([CATEGORY])
        …
;

[CATEGORY]
        (category [CULTURE])
        (category [SPORTS])
        …
;

[CULTURE ]
        (culture)
;

[SPORTS]
        (sports)
;
```

Figure 17. Excerpt grammar of the news module.

This convention is not mandatory for the use of the translation service as it works with any correct grammar, but revealed essential for the development of AALFred.

3) *AALFred semantic grammars*

Following the conventions presented in the previous section, the different teams of developers created the grammars for the spoken interaction of their modules. After, all the fragments were added together, resulting in a large grammar with a total of more than 250 rules, which were automatically translated for each language.

4) *On the translated grammars*

As native speakers of each of the languages to which grammars were translated made a manual revision, some insight on the quality and usefulness of the translation process can be obtained by comparing the automatically obtained grammars with the revised versions.

Fig. 18 shows the number of lines for the grammar in each language before and after the manual revision. The graphic shows that automatic translation inserts more sentences, with the same meaning, to the original (all languages have more lines that English). Portuguese and French revisers added more sentences in addition to those already in automatic translation and Polish and Hungarian have deleted some.

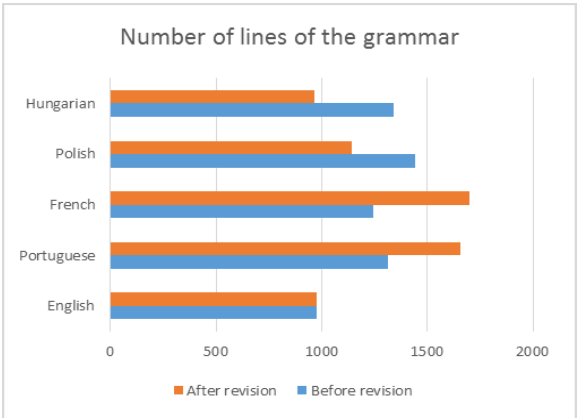

Figure 18. Number of lines for the grammar in each language before and after the manual revision.

In order to investigate in more detail, as human reviewers had the freedom to add and delete in any position, an automatic alignment process was applied to the pair of automatic translated and revised grammars, based on the number of changed characters and using the Levenshtein distance. A threshold of 20% of changes was adopted to decide between 3 classes: edited, deleted, and added. Also a 2 columns HTML output of the process results was created to allow manual inspection of the results. This HTML was used to define the mentioned threshold.

The result of this analysis is presented in Fig. 19. The graphic shows the number of new lines, number of lines that remained unaltered, number of lines that had some minor edition and number of lines that were removed.

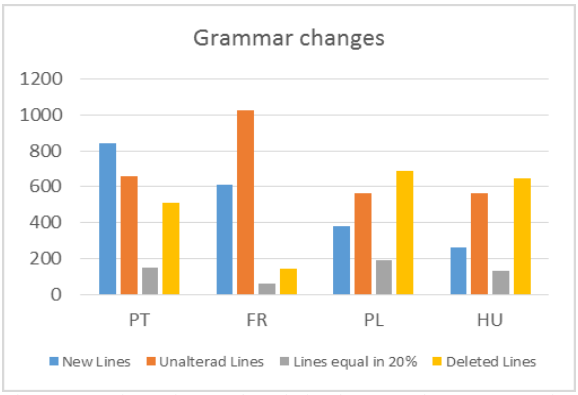

Figure 19. Information on the relation between the grammars obtained by automatic translation and their revised versions.

French was the language where more lines were accepted by the human reviewer. Also, while for French and Portuguese more than 500 lines were added, for Hungarian and Polish there were more deleted lines than new lines. This results combined suggest two different groups of system use. For one group the base grammar obtained by translation was used as a basis for extension; for the other to create a more correct one.

As the reviewers also reported that the revision process took a reasonable amount of time, all added, the results from the creation and translation process for AALFred were positive. Also, the PaeLife consortium decided for the usage of the grammars obtained and of the service to support SLU in the field trials in France, Hungary, and Poland. In the scope of another project, the Portuguese national project AAL4ALL [29], the Portuguese version of AALFred was also subject to field trials using the service described in this paper. The results of these evaluations will be the subject of forthcoming publications, some in preparation.

VI. Discussion

The results of the translation and manual revision of AALFred's grammars show differences regarding the target language used for translation. In a group of languages, human reviewers mostly added new patterns and made some corrections, enriching the grammar. For other languages, revision consisted mainly in removing incorrect patterns.

While these results could at first point to a very weak usefulness of the service, the feedback from developers and project partners point in a very different direction. In fact, for the application envisioned for the service, the many additions made were potentiated by the fact that an initial grammar already exists and reviewers could look for missing cases. Also some of the additions are enrichments to the original English grammar, adding patterns not initially included by the developers.

The fact that deletion of incorrect patterns was higher for languages such as Hungarian or Polish can eventually be related to a higher degree of complexity of the translation from English to those languages and/or lower performance of the translator.

For both groups of languages, the initial grammar produced by translation revealed itself as very useful.

VII. Conclusions and Future Work

Supporting multiple languages in spoken language interaction with machines is a complex task. In this paper is proposed the use of automatic translation applied to the generation of grammars for a set of target languages having as a basis a semantic grammar in English created by the application developers. Also, 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 the speech recognition and SLU modules in different languages by automatic translation of an existing grammar (in English). A first prototype has been implemented, tested, and adopted in a Personal Assistant, AALFred.

Several examples of the service capabilities are presented. These examples are complemented with information regarding how the service is being used to support both development and real usage of AALFred.

Future developments should explore the use of multiple translation services, increasing the probability of having, in the set of translated sentences, the correct ones. Also, the service has potential to be used in new applications using this or other sets of languages.

Acknowledgments

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] A. Teixeira, P. Francisco, N. Almeida, C. Pereira, and S. Silva, "Services to Support Use and Development of Speech Input for Multilingual Multimodal Applications for Mobile Scenarios," in *The Ninth International Conference on Internet and Web Applications and Services (ICIW 2014),Track WSSA*

- *Web Services-based Systems and Applications*, 2014, pp. 41–46.

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

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

[4] C. Munteanu, M. Jones, S. Oviatt, S. Brewster, G. Penn, S. Whittaker, N. Rajput, and A. Nanavati, "We need to talk: HCI and the delicate topic of spoken language interaction," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, 2013, pp. 2459–2464.

[5] A. Teixeira, F. Ferreira, N. Almeida, A. Rosa, J. Casimiro, S. Silva, A. Queirós, and A. Oliveira, "Multimodality and Adaptation for an Enhanced Mobile Medication Assistant for the Elderly," in *Proc. Third Mobile Accessibility Workshop (MOBACC), CHI 2013*, 2013.

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

[7] A. Teixeira, A. Hämäläinen, J. Avelar, N. Almeida, G. Németh, T. Fegyó, C. Zainkó, T. Csapó, B. Tóth, A. Oliveira, and M. S. Dias, "Speech-Centric Multimodal Interaction for Easy-To-Access Online Services : A Personal Life Assistant for the Elderly," in *Proc. DSAI 2013, Procedia Computer Science*, 2013, pp. 389–397.

[8] J. H. Christensen, "Using RESTful web-services and cloud computing to create next generation mobile applications," in *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, 2009, pp. 627–634.

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

[10] "iOS - Siri." [Online]. Available: http://www.apple.com/ios/siri/. [Accessed: 14-Mar-2015].

[11] "Meet Cortana to Windows Phone." [Online]. Available: http://www.windowsphone.com/en-us/how-to/wp8/cortana/meet-cortana. [Accessed: 14-Mar-2015].

[12] "PaeLife: Personal Assistant to Enhance the Social Life of Seniors." [Online]. Available: http://www.microsoft.com/portugal/mldc/paelife/. [Accessed: 14-Mar-2015].

[13] A. Hämäläinen, A. J. S. Teixeira, N. Almeida, H. Meinedo, T. Fegyó, and M. S. Dias, "Multilingual speech recognition for the elderly: The AALFred personal life assistant," in *Proc. DSAI 2015, Procedia Computer Science*, 2015.

[14] "Voice Search." [Online]. Available: http://www.google.com/insidesearch/features/voicesearch/index-chrome.html. [Accessed: 14-Mar-2015].

[15] 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.

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

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

[18] A. Hunt and S. McGlashan, "Speech Recognition Grammar Specification Version 1.0." [Online]. Available: http://www.w3.org/TR/speech-grammar/. [Accessed: 14-Mar-2015].

[19] G. Tur and R. De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[20] R. De Mori, F. Bechet, D. Hakkani-Tur, M. McTear, G. Riccardi, and G. Tur, "Spoken language understanding," *Signal Process. Mag. IEEE*, vol. 25, no. 3, pp. 50–58, 2008.

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

[22] A. Teixeira, N. Almeida, C. Pereira, and M. O. 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*, 2013.

[23] N. Almeida and A. Teixeira, "Enhanced interaction for the Elderly supported by the W3C Multimodal Architecture," in *5ª Conferência Nacional sobre Interacção*, 2013.

[24] W. Ward, "Understanding spontaneous speech: the Phoenix system," in *International Conference on Acoustics, Speech, and Signal Processing, 1991. ICASSP-91.*, 1991, vol. 1, pp. 365–367.

[25] "Phoenix Grammar Reference - Olympus." [Online]. Available: http://wiki.speech.cs.cmu.edu/olympus/index.php/Phoenix_Grammar_Reference. [Accessed: 14-Mar-2015].

[26] "Word Alignment Information from the API." [Online]. Available: http://msdn.microsoft.com/en-us/library/dn198370.aspx. [Accessed: 14-Mar-2015].

[27] H. Bunt, J. Alexandersson, J. Carletta, J.-W. Choe, A. C. Fang, K. Hasida, K. Lee, V. Petukhova, A. Popescu-Belis, L. Romary, C. Soria, and D. Traum, "Towards an ISO Standard for Dialogue Act Annotation," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 2010.

[28] S. Young, "CUED standard dialogue acts," 2009.

[29] "AAL4ALL | Ambient Assisted Living For All." [Online]. Available: http://www.aal4all.org/?lang=en. [Accessed: 14-Mar-2015].