Automatic speech recognition for assistive technology devices

A P Harvey[1], R J McCrindle[2], K. Lundqvist[3] and P Parslow[4]

[1,2,3,4]School of Systems Engineering, University of Reading,
Whiteknights, Reading, Berkshire, UK

[1]*a.p.harvey@rdg.ac.uk,* [2]*r.j.mccrindle@rdg.ac.uk;* [3]*k.o.lundqvist@rdg.ac.uk;* [4]*p.parslow@rdg.ac.uk*

[1,2,3,4]*www.reading.ac.uk/sse*

ABSTRACT

Speech offers great potential as a mode of interaction with devices to control our environment, support our work or assist us with tasks of daily living, however, to date the level to which this has been universally achieved and exploited has not matched its potential. Automatic Speech Recognition (ASR) is the process of interpretation of human speech by a machine. This may take two forms; continuous speech, as with human to human interaction or dictation, and discrete speech, such as commands issued to a device. ASR in the 'ENABLE' project uses discrete utterances to allow navigation of the user interface on a wrist worn device, control of the associated ECS (Environmental Control System) components as well as the ability to provide feedback for long term conditions using speech alone; features widely requested by users with a range of disabilities as well as by others for general ease of use. The aim of this paper is to explain the rationale and process behind the development of the ASR for the ENABLE device.

1. INTRODUCTION

Automatic Speech Recognition (ASR) is the process of interpretation of human speech by a machine. This may take two forms; continuous speech, as with human to human interaction or dictation, and discrete speech, such as commands issued to a device. Speech offers great potential as a mode of interaction with devices to control our environment, support our work or assist us with tasks of daily living. Recognition of continuous speech is primarily used to compose a document with dictated speech replacing the need to use a mouse and keyboard, whilst discrete speech recognition is most commonly used for command based activities such as those associated with automated call centres and non safety critical military applications. However, despite these examples, integration of ASR into systems that could benefit from its application has not been universally achieved and its exploitation despite having considerable potential and to date has fallen far short of expectations.

ENABLE is an EU project aimed at developing a wearable device that can be used both within and outside of the home to support older people in their daily lives and which can monitor their health status, detect potential problems, provide activity reminders and offer communication and alarm services. The system is built round a dual platform of mobile phone and wrist unit, to which are added modular capabilities for (1) alarm functions, (2) control of appliances/other devices around the home, (3) identification of the user's location, and (4) health monitoring. The project also addresses mobile phone accessibility by providing an accessible user interface extension, using the wrist unit and speech guidance.

ASR in the ENABLE project uses discrete utterances to allow navigation of the user interface on a wrist worn device, control of the associated ECS (Environmental Control System) components as well as the ability to provide feedback for long term conditions monitoring using speech alone; features widely requested by users with a range of disabilities as well as by others for general ease of use. In total 132 commands have to be recognised in order to fully operate the wrist unit (WU) including commands such as 'back', 'select, 'up', 'down', 'light-on', 'blinds up', 'play', 'rewind', etc. A full list of the commands is given in Figure 1 at the end of the paper. The aim of this paper is to explain the rationale and process behind the development of an ASR system for the ENABLE device and to discuss the key challenges associated with implementing ASR on a wearable assistive technology device.

2. EXISTING ASR SYSTEMS

2.1 Speech Recognition Engines

Many speech recognition engines/systems are available on commercial and open-source licences such as Google Voice Search [Google, 2010], Nuance (Philips) [Nuance, 2010], LumenVox [Lumenvox, 2010], Loquendo [Loquendo, 2010], Cambridge HTK [HTK, 2010], simon [Speechtotext, 2010] and CMU Sphinx [CMU, 2010] each having different strengths, weaknesses and capabilities that make them appropriate for different tasks. A number of speech research groups across Europe including the ESAT Speech Group, K.U.Leuven, Belgium [ESAT, 2010], the Wire Communications Laboratory, University of Patras, Greece [Patras, 2010], and the Speech Processing Group, Brno University of Technology, Czech Republic [Brno, 2010] are also developing new ASR engines for specific localisations and improvement of existing engines for particular languages.

2.2 Projects Incorporating ASR

A range of projects incorporating ASR have been/are being undertaken all of which whilst interesting, have a different application area/focus, or which only partially address the combination of aspects of ASR that are required by the ENABLE system. These projects include: Luna [Luna, 2009], PAST (exPeriencing Archaeology across Space and Time) [MJC2, 2007], EASAIER (Enabling Access to Sound Archives through Integration, Enrichment and Retrieval) [Reiss, 2008], MAESTRO [Rivlin et al, 2000], Companions [Cavazza, 2010)], SynFace [SynFace, 2004], DICTATE [Nuance, 2010], Stardust [Hawley, 2003], Inspire [Inspire, 2004], Speecon [Speecon, 2000], IDAS [Patras, 2010], Amigo [Philips, 2007], Agent-DYSL [Agent-Dysl, 2009], Vital Mind [Cognifit, 2005] and HERMES [Jiang et al, 2009],

3. CHALLENGES ASSOCIATED WITH ASR

The numerous complexities associated with developing ASR systems have meant that traditional ASR systems have frequently not been fully appreciated by the end-user; have not been adapted to meet user requirements; must be specifically trained to an individuals' voice; or simply have not been able to recognise words with enough accuracy. For example:

- Differences in voice characteristics exist between users, compounded by different accents/dialects and sometimes entirely different pronunciations.
- Similarity of certain words and confusion of sounds such as 'n' and 'm'.
- Short words are more difficult to recognise as there is much less data to process and context within which to place it, e.g. "up", "down" and other monosyllabic words.

This situation is further compounded if all or some users of the system are elderly people or they have some degree of impaired speech, e.g.:

- There may be less consistency in the voice of an elderly person due to tiredness.
- Medical conditions such as stroke, dysarthria, or breathlessness may affect speech.

There may be high levels of background noise e.g. due to the television or other people talking

4. ENABLE APPROACH TO DEVELOPMENT

4.1 Overall Approach to the ASR Development Process

Effective development is frequently a mix of integration and implementation – and this is the approach we have adopted in ENABLE. Indeed this approach is commonly adopted across many integrated research and development projects that involve ASR, the key decision being which engine to use (e.g. the proprietary systems of Loquendo, Nuance or Philips vs. open source systems such as Sphinx or simon). This decision to utilise a speech engine rather than develop our own as part of the ENABLE project was made for a number of reasons:

1. Many person-years of effort have been input into the development of the Sphinx (or similar) engine and to replicate this would not have added value to the project but would have consumed additional resources.

2. In light of the project requirements it was far more important to develop the ASR models, grammars, functions and extensions required for the ASR to interface to and be integrated with the ENABLE system than to create an untested engine.
3. It makes sense to make use of the features such as the noise cancelling facility and automatic level control of the Sphinx engine and the audio chip embedded in the wrist unit.
4. It was possible to take advantage of the robustness of an engine that has been maturing over a number of years.

Following a period of initial investigation and prototyping, PocketSphinx, a state of the art open source speech recognition engine was selected as the foundation for the ENABLE ASR development, due to the reliability of its engine and its inclusion of various algorithms to compensate for noise and other external factors affecting the reliability of the recognition. However, 'out-of-the-box', PocketSphinx provides only a recognition engine and an audio model for North American English (and Mandarin in the latest version). A model in this sense comprises only phonemes and a statistical representation of each; it does not contain a corpus, dictionary or grammar all of which had to be created. A new model for each required language also had to be developed (including U.K. English).

Creating a new model for PocketSphinx to allow it to recognise words in other languages involved mapping phonemes and gathering audio data for each necessary language. Phonemes are specific to each language, even between dialects. For instance, U.K. English has 5 more phonemes than U.S. English. Once the audio data for each language was gathered, each recording was separated into its constituent phonemes for the purposes of training. Once this had been achieved, the recordings were processed to form a Hidden Markov Model (HMM). Further phonemic analysis of the command corpora was then performed to enhance the recognition rate for different English dialects and accents. From the work undertaken during this period of investigation, additional alternative pronunciations were added to enhance the models' performance characteristics. Vowels (and vowel sounds) were taken into particular account, alongside consonants with similar sounds, such as 'n' and 'm', etc.

Additionally, PocketSphinx provided only a simplistic API (Application Programming Interface) to interface with. A wrapper around this had to be developed for the ENABLE ASR system, to allow the Wrist Unit (WU) to start and stop the service to conserve battery life, pass memory locations to recordings of speech in order to return recognised utterances and provide a mechanism to 'jump' between branches in the finite state grammar tree (Figure 1). Essentially, the grammar is broken down into finite states where each non-terminal command 'leads' to a further set of commands. This means that each command can only be uttered in its intended position in the sequence of commands. Each position in the menu allows only a certain number of commands to be uttered, meaning that the potential for misrecognition is reduced to a proportion of the number of linked commands (see Figure 1 at the end of the paper). *This is context-sensitive ASR.*

4.2 Stages in the ASR Development Process

In order to implement the above approach a number of stages of development were undertaken. These are described below and summarised in Figure 2.

4.2.1 Stage 1 – Rapid Application Development. Initial work on the ASR system produced a speech recognition prototype based on Sphinx 4, an open source speech recognition engine

4.2.2 Stage 2 – Selection of Approach. The decision to use an available engine was made because of the reliability of current engines. Various techniques in speech recognition have been adopted as the standard, and attempting to improve on these in a relatively short period of time would likely prove unfruitful. However, as detailed above, using an engine does not mean that the difficulties are all overcome. Models, grammars, corpora, etc. must be produced for the engine, which is where the state-of-the-art developments for the ENABLE Wrist Unit have focussed.

4.2.3 Stage 3 – Selection of Final Platform. The decision to switch development to 'PocketSphinx' was made after assessing the 'Sphinx' engine. It was found to be robust, but far too heavyweight to run on an embedded device. A number of speech recognition engines were evaluated, considering their expected performance on low-power hardware, the nature of their associated licences, cost and system requirements. PocketSphinx was chosen based on its performance, appropriate system requirements, its similarity to the already developed Sphinx prototype and the permissive nature of its licence.

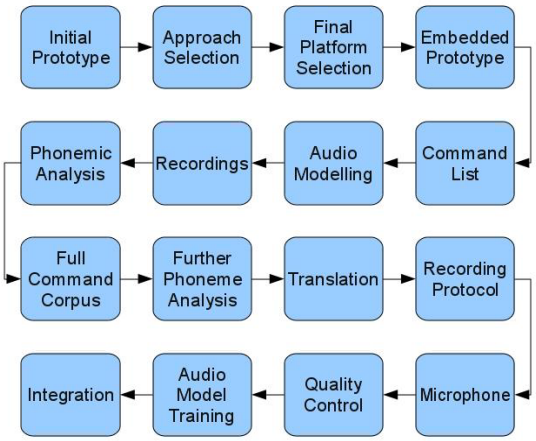

Figure 1. *Development Stages for the ENABLE ASR System.*

4.2.4 Stage 4 – Prototype Development. A preliminary software prototype was produced to develop, test, analyse and modify underlying audio and language models using PocketSphinx. A new language (audio) model based on a Finite State Grammar has been developed and integrated into the PocketSphinx based toolset. This Finite state grammar forms the basis of the menu structure within the ASR system; it can be (and has been) expanded to cover a wider range of commands or modified to produce a new structure based on the requirements of the user and the menu in the user interface of the ENABLE device.

4.2.5 Stage 5 – Command List. A corpus of possible commands was produced for the PocketSphinx prototype. This corpus was based on the available commands/functions used in previous demonstrations of the ENABLE prototype. Each command was taken from the UI (User Interface) menu structure and added to the corpus, and its place within the structure recorded in the finite state grammar. Each word was then broken down into its constituent phonemes to form a command list that PocketSphinx is able to use to identify words. This corpus was subsequently translated into Czech, Dutch, German and Greek, and initial phonemic analysis of the commands was executed to comprise command lists similar to the format used for the English list, appropriate for the PocketSphinx engine.

4.2.6 Stage 6 – Modelling. This preliminary model was used to test audio modelling using existing corpora to identify problems between different words. A basic grammar was used, expecting only a single word at each utterance. This allowed words from the initial set of demonstrated commands to be verified against each other, and words which may be easily misrecognised as other words to be easily identified. Where such confusion was likely, alternative words were identified, and then analysed for the likelihood of the words existing at the same point in the grammar (i.e. could these words be options as part of the same sub-menu).

4.2.7 Stage 7 – Recordings. A series of audio recordings were taken for the initial command set in English. This included people of varying backgrounds, accents, ages and genders. These recordings have been used in a number of ways; to test the performance and reliability of the system, to further enhance the audio models used by the system to enhance its reliability and form a basis for an audio recording protocol to capture further audio samples in all partner languages. Further recordings of this command set were made at the pilot sites in Czech, Danish, Dutch, German and Greek. Similarly, these recordings were used to produce a basic model for each language, utilising the initial command corpus.

4.2.8 Stage 8 – Improvements. Further phonemic analysis of the command corpora was performed to enhance the recognition rate for different English dialects and accents. From the work undertaken during this period of investigation, additional alternative pronunciations were added to enhance the models' performance characteristics. Vowels (and vowel sounds) were taken into particular account, alongside consonants with similar sounds, such as 'N' and 'M', etc. Adjustments were made to the models for the other languages based on the results of the tests performed using the obtained audio samples. These improvements were tested with these recordings to gauge the level of improvement, or readjust where necessary.

4.2.9 Stage 9 – Full Corpus. After testing of the prototype, and confirmation of various techniques for the modelling of languages other than English, the full command corpus for the ENABLE device was defined. This command list was formed from input from various partners, including an assessment of all of the underlying functions of the device. The fundamental objective for development of the corpora was to ensure that the ENABLE device ASR function could deliver all command features in the voice control module.

Other words, such as locations, were added to the device as potential shortcuts, allowing various menu items to be accessed from other areas in the menu structure.

4.2.10 Stage 10 – Detailed Phoneme Analysis. A detailed analysis of the English command list was performed to identify the phonemes comprising each word. This determined which phonemes were appropriate for the commands specified, and analysed the words to incorporate different sounds and different interpretations of the audio recordings, depending on the quality (background noise, etc.) to ensure the best possible matches for each. The analysis also included an investigation of the positions of the vowels in the words, and the addition of similar sounding vowel phonemes as duplicate definitions of those words to allow for different accents, dialects and any degradation of voice.

4.2.11 Stage 11 – Translations. Translations into Czech, Dutch, French, German and Greek were completed and a programme of work to analyse the full command list was undertaken and the resulting constituent phonemes were correlated with the English equivalent. The lists of phonemes produced, similar to those used in the prototype, were used alongside the current audio models and tested.

4.2.12 Stage 12 – Recording Protocol. A new audio recording protocol, specific to the ENABLE project, was defined, specifying the process to be followed when taking audio samples. It specifies that participants select (although the rest of the process is applicable to any voice recording for any participants), how the words in a command list should be recorded together/individually, including special instructions for alternative pronunciations and synonyms, the naming conventions to be used for files and potential methods of delivery of the files. This protocol was followed locally at Reading to test it, and to obtain recordings in English before passing it to other Partners.

4.2.13 Stage 13 – Microphone. All recordings taken were using the same model of microphone included in the ENABLE device, albeit attached to a PC or other recording device. This is important, as different microphones have different dynamic ranges, and thus produce different sounds to other similar microphones. Due to the nature of these and other characteristics of potential microphones, it is essential for a functioning ASR system to utilise a suitable microphone and thus obtain a consistent signal. The microphone chosen is a highly directional model, meaning that most of the recorded sound should be from the direction of the participant. This also contributes to the success of the ASR system by not recording a substantial proportion of the background noise in general operation.

4.2.14 Stage 14 – Quality Control. Returned recordings were examined for quality, ensuring that there were no interruptions from loud noises in the background, labels were correct and audio levels were sufficient. Where possible, these were corrected with software so as not to distort the recording, and/or re-recorded. Using the lists of phonemes produced for each language, each recording was transcribed into its constituent phonemes. Large periods of silence were removed, along with coughs, sneezes, etc. With the recordings, adjustments were made to the list of phonemes where appropriate, to ensure the best fit for training the models.

4.2.15 Stage 15 – Training the Models. Each recording and accompanying transcription was used to train a language dependent model. A tool for this is provided with Sphinx. The trained models were then tested, and further phonemic analysis was performed against the recordings in all languages. Scripts were defined for testing against the existing models to semi-automate the process. Adjustments were made to the audio models and phoneme compositions as and where necessary. Adaptation and tweaking of the audio models was performed as necessary after testing.

4.2.16 Stage 16 – Integration. In preparation for integration with the other features designed and developed for the wrist device, cross-compilation of modules and libraries for use on embedded hardware was completed for all components necessary for ASR. After further checks and verification of the compilation process for non-x86-based hardware, the process of integrating the ASR system into the build-tree began during an 'Integration Week'.

Consultation with the relevant technical partners regarding the basis for the integration of PocketSphinx was established and commenced with writing a 'Make' file to extract and build the libraries, before inserting them into the correct locations within the build-tree. This file was linked back into the main build-tree so that the process of compiling the entire tree would include the ASR libraries.

A plug-in for the existing user interface was created, which loads the ASR system, passes it details about the language and grammar to use for the session, passes the audio recordings for recognition and manages the returned strings to the user interface. This plug-in is the core of the integration work for ASR and manages all

aspects of the ASR subsystem. Initial work on the ASR system produced a speech recognition prototype based on Sphinx 4, an open source speech recognition engine

The ASR requires a different integration strategy than the other ENABLE services such as the falls detection and environmental control system (ECS) which can be effectively implemented as stand-alone plug-ins. As each ASR commend has to evoke a different response in the system each command has to be associated with an action both in the main navigation part of the code and for each action within each of the sub-sections; thus making integration a more complex issue.

5. RESULTING ASR FEATURES IN THE ENABLE SYSTEM

The following key state-of-the-art features have been developed as part of the ENABLE ASR system:

- Development of a new language (audio) model based on a Finite State Grammar which forms the basis of the menu structure (130 words in total) within the ASR system and which is extensible to meet the requirements of the user, or for inclusion of future wrist unit functionalities. The ENABLE ASR system utilises PocketSphinx, a state of the art open source speech recognition engine. On top of this engine, the speech models, in Hidden Markov Model (HMM) form, are passed to the engine for processing. This provides a basis for the individual phonemes to be recognised on a probability basis to form words. These words are included in the corpus for each language, as described above. The words are recognised by forming Markov Chains in the model; these are the most likely 'paths' through the model that the spoken word may take.
- Creation of a multi-lingual command corpus (English, Czech, Dutch, German and Greek). A series of language and speech models have been produced for the each of the languages involved in the WP11 pilot trials. These are 'compiled' versions of speech recordings from EPs for a particular language, forming a HMM for each. A corpus of all possible words which may be spoken to the device has been defined for each language. These provide a definition of each of the phonemes (or potential phonemes where more than one is possible in different dialects) which form each word.
- Development of a context-sensitive mechanism to lessen the likelihood of misrecognition of words (through structured grammar). Each wrist device can be configured with a menu structure specific to the ECS configuration related to the user. The specific menu structure will generate an ASR grammar specific to that configuration. The grammar is broken down into finite states where each non-terminal command 'leads' to a further set of commands. This means that each command can only be uttered in its intended position in the sequence of commands. Each position in the menu allows only a certain number of commands to be uttered; meaning that the potential for misrecognition is reduced to a proportion of the number of linked commands.
- Minimisation of ambient noise by including with each language's audio model recordings of various imperfections in speech input, such as pops, hisses and crackles and other background noises so that the speech recognition engine can detect such noise and ignore it. Included with each language's audio model are recordings of various imperfections in speech input, such as pops, hisses and crackles and other background noise. This is designed to enable the speech recognition engine to detect such noise and ignore it. Combined with a specifically chosen directional microphone, and a software gate and compressor system to normalise the speech input (providing a threshold), the ASR system takes input from a robust speech input device.
- Development of an ASR system capable of running on embedded hardware (without the need for an external server to handle the model processing). A major constraint relating to the design of the architecture was the necessity to integrate the ASR into an existing system. This is coupled with the embedded nature of the device (slower, non-x86-based hardware) and the modular nature of the ECS system; all features of ECS have the ability to be controlled by the ASR, which uses a fixed grammar rather than the ever changing structure of the ECS menu, which depends on a number of factors, including location
- Creation of an ASR system that will work in all partner languages (English, Czech, Dutch, German and Greek) and which can be extended to include all languages/dialects.
- Speaker independence for majority of older people removing the need for voice training. As the models are created from a large set of transcribed audio, the ASR system is speaker-independent for the majority of elderly users. This means that the EP will not have to train the ASR system to their

own voices, which has been identified as a source of dissatisfaction and problems with other speech recognition systems

- Ability to develop speaker dependent language models based on an elderly person's long term condition (e.g. stroke). A new language model based on an individual's speech patterns could be developed in a similar manner to developing a new model for an additional language/dialect.
- Working ASR system on the wrist unit within a 2 second processing limit

Whilst each of the above features in itself is not totally unique, as other ASR systems can demonstrate one or more of the above features, the key innovation in ENABLE comes from having implemented ALL of the above features in a single device and on non-proprietary software such that "the whole is greater than the sum of its parts."

In contrast, many speech recognition systems are specific to a small set of languages, *are* limited to equipment with large amounts of processing power (and thus not portable), use natural language processing to attempt to recognise entire sentences or dictated/continuous speech or use the power of a server/multiple servers to process the speech remotely from the mobile device. This incurs high communication costs and increases the time taken to process each word (due to communication delays, server availability, load, etc.) and limits the device to areas within communication range.

Additionally, although some ASR based devices do work very effectively, many other current mobile speech recognition devices are considered to be inadequate; this has been highlighted in general public and internet opinion and in a number of conferences. Training is required for general speech recognition software on mobile devices*;* a significant challenge for the ENABLE ASR system was to produce a speaker independent speech recognition system, capable of recognition on a mobile device (wrist unit) with a suitable period of time. Most current state of the art speech recognition systems require training, powerful (fast) hardware and even then do not produce impressive results.

After all '*if ASR was easy to do it would be everywhere*'.

6. TESTING AND RESULTS

By adopting the context sensitive approach to ASR it reduces the number of words that need to be recognised at any one time and hence recognition rate should be improved. For example: Total states for each command in the ENABLE system without context sensitive ASR is 130. With Context sensitive ASR the number of states from which to select the correct word are considerably reduced (see Figure 1 and Table 1).

Table 1. *Examples of Context Sensitive ASR*

Case 1: Menu -> Remote Control -> Living Room -> Television -> Volume Up	**Case 2: Menu -> Remote Control -> Bedroom -> Lights Brighter**
Possible potential states: Menu: 9 Remote Control: 5 Living Room: 7 Television: 6 Volume up: 8	Possible potential states: Menu: 9 Remote Control:5 Bedroom: 7 Lights Brighter: 8

Word Recognition Rates (WRR) through this approach for each language model when tested under laboratory conditions are:

Table 2. *WRR for different language models in the ENABLE system.*

English 92.4%
German 88.6%
Dutch 75.8%
Greek 72.7%
Czech 69.7%

The project is currently exploring several potential explanations for why words in some languages are recognised more easily than others, for example:

- The recordings taken could be better for some languages than others
- The nature of the language in question has different characteristics, so German, for instance, has some quite hard sounding consonants that are easy for Sphinx to identify, whereas the audio profile for, say, Czech is much softer.
- Some languages tend to have phonemes that are nearly indistinguishable from each other to anyone but a native speaker, (think N and M), so a computer is going to have similar difficulties to foreign speakers.

Field trials with users of the ENABLE wrist unit device and the ASR system are being conducted in the UK, Belgium, Austria and the Czech Republic during July and August 2010.

7. DISCUSSION AND CONCLUSIONS

Automatic Speech Recognition is a highly complex task and is not quick and easy to implement, even with the assistance of a speech engine. The ENABLE ASR system supports multiple languages, runs reliably on a mobile device, and uses direct command words and short cuts expected of an easy to use device. The system, when delivered to a typical user, will work "out of the box" and not require any training. The system is targeted at elderly people, whose voices have been used to produce the current speech models. This will be common across all of the defined languages.

The framework for further language development and for impeded speech recognition has also been developed. It is also planned to use the ASR components of the ENABLE system as the basis for a number of further research projects linked to speech recognition; web-based browsing; and ASR for the deaf community; as well as ASR recognition in different contexts. We will also make the new language recognition models available to the wider ASR community (UK, EU and beyond)

Acknowledgements: The ENABLE Project (2007-2010) is funded in part by the European Commission in the 6th Framework Programme (project number: 045 563). Partners are: AT: fortec - Vienna University of Technology – IS, KII - Kompetenznetzwerk Informationstechnologie zur Förderung der Integration von Menschen mit Behinderungen; ES: ARTEC; UK: Docobo Ltd, Cardionetics Ltd., Reading University; CZ: Zivot 90; SP: Code Factory Ltd.; GR: E-Isotis, BE: Vzw Cassiers Wzc; Website *http://www.enable-project.eu*/.

8. REFERENCES

Agent-DYSL (2009), Accomodative Intelligent Educational Interfaces for Dyslexic Learners, http://www.agent-dysl.eu/

Brno University of Technology (2010), Speech Processing Group**,** http://speech.fit.vutbr.cz/

M Cavazza, Companions, Intelligent, persistent, Personalised, Multimodal Interface to the Internet, http://www.companions-project.org/

CMU (2010), Sphinx, http://cmusphinx.sourceforge.net/

Cognifit Limited (2005), Vital Mind, http://cordis.europa.eu/fetch?CALLER=PROJ_ICT&ACTION=D&CAT=PROJ&RCN=85774

ESAT (2010), ESAT PSI Speech Group, http://www.esat.kuleuven.be/psi/spraak/

Google (2010), Google mobile app, http://www.google.com/mobile/google-mobile-app/

M Hawley (2003), STARDUST - Speech Training and Recognition for Dysarthric Users of Assistive Technology, http://www.fastuk.org/research/projview.php?id=216

HTK (2010), Hidden markov model toolkit, http://htk.eng.cam.ac.uk/

INSPIRE (2004), INfotainment management with SPeech interaction via REmote-microphones and telephone interfaces, http://www.ist-world.org/ProjectDetails.aspx?ProjectId=578d8793b1404521ab29a62b1ae55108

J Jiang, A Geven, B Zhang (2009), Hermes - Computer-Aided Memory Management Via Intelligent Computations http://www.springerlink.com/content/t6u276605n5h6v1h/

Loquendo (2010), Loquendo ASR, http://www.loquendo.com/

Lumenox (2010), Lumenox speech engine, http://www.lumenvox.com/

Luna (2009), Spoken language understanding in multilingual communication systems, *http://www.ist-luna.eu/index.htm*

MJC2 Limited (2007), PAST, experiencing archaeology through space and time, http://cordis.europa.eu/fetch?CALLER=PROJ_ICT&ACTION=D&CAT=PROJ&RCN=52653

Nuance (2010), Nuance speech solutions, http://www.nuance.co.uk/

Nuance (2010), Dictate Medical, HTTP://www.dragon-medical-transcription.com/

Patras (2010), Wire Communications Laboratory, http://www.wcl.ece.upatras.gr/

Patras (2010), A Dialogue System for Telephone-based Services, http://www.wcl.ece.upatras.gr/ai

Philips (2007), Amigo, Ambient Intelligence for the networked home environment, http://www.hitech-projects.com/euprojects/amigo/project_information.htm

J Reiss (2008) EASAIER, prototype on speech and music retrieval systems with vocal query interface http://www.elec.qmul.ac.uk/easaier/publicdeliverables/d3_2_speechmusicprototype_v19%20(2).pdf

Z Rivlin, D Appelt, R Bolles et al (2000), Maestro, conductor of multimedia technologies, http://www-speech.sri.com/projects/sieve/Maestro.pdf

SPEECON (2000), Speech Driven Interfaces for Consumer Devices, http://www.speechdat.org/speecon/index.html

Speechtotext (2010), simon Speech Recognition software, http://sourceforge.net/projects/speech2text/

Synface (2004), Synface, http://www.speech.kth.se/synface/index.htm

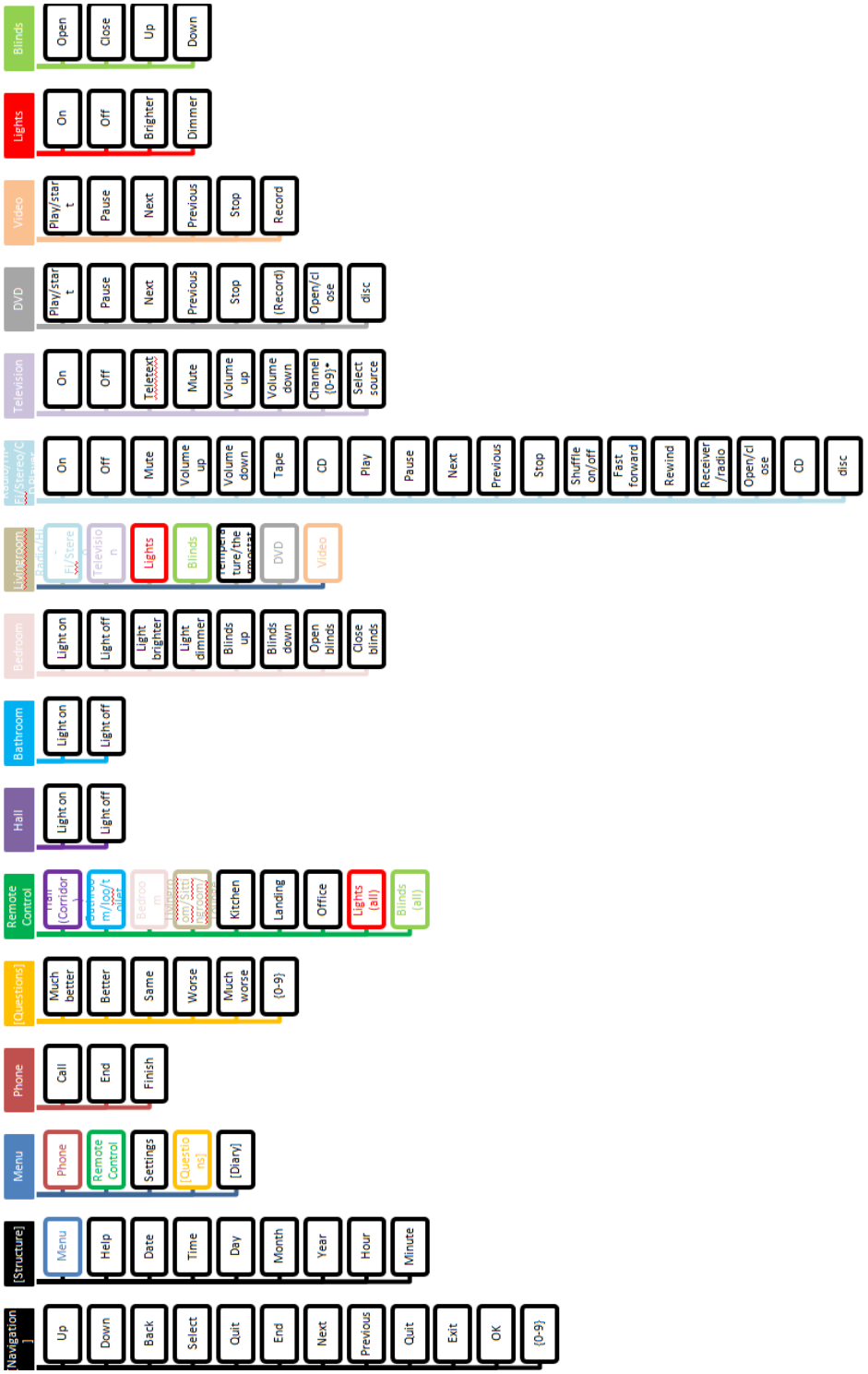

Figure 1. *ASR Command Structure Implemented on the WU.*

Proc. 8th Intl Conf. Disability, Virtual Reality & Associated Technologies
Viña del Mar/Valparaíso, Chile, 31 Aug. – 2 Sept. 2010