

ECHORD

Technical Report D3.1:

Mid-term Home Service Prototype and Demonstrator

***E. Bastianelli*, F. Giacomelli**,
L. locchi*, N. M. Manes**,
D. Nardi*, V. Perera****

Proposal full title: **SPEAKY for Robots**

Proposal Acronym: **S4R**

Name of the Coordinating Person: **Daniele Nardi**

Institution of the Coordinating Person (*): **Sapienza Università di Roma, Dipartimento di Ingegneria Informatica, Automatica e Gestionale**

Other participating institution (**): **Mediavoice S.r.l.**

Preface

SPEAKY for Robots (S4R) aims at fostering the definition and deployment of voice user interfaces (VUIs) in robotic applications where human-robot interaction is required. More in depth, S4R promotes speech technologies transfer towards manufacturing processes, to provide semi-automatic speech-based interface development for robotic platforms. This in turn will boost up the robot presence in manifold activities, by supporting a natural interaction with humans.

S4R specific goal is a novel **Robotic Voice Development Kit** (RVDK), namely a framework that supports robotic developers in designing voice user interfaces with little effort. RVDK is conceived as an interactive environment aiding designers to define the voice interface according to the desired application requirements; hence, it is adaptable to different application fields. In order to design and implement the RVDK, state of the art solutions about lexical vocabularies and knowledge representation to capture the semantics of the domain, and natural language processing technologies will be integrated to build the input for the speech processing system SPEAKY, that is currently commercialized by Mediavoice partner of the consortium.

S4R experiment targets two possible application domains for RVDK. The first scenario deals with **home services**, where a user controls a humanoid robot through a voice user interface within a domestic environment. The second scenario consists of an **outdoor robotic surveillance**, where the user is controlling the action of a wheeled robot capable of navigating in rough terrain. The goal of the experiments is to assess the performance of the voice interface implemented though RVDK. This project is thus a **joint enabling technology development** effort, whose scope falls within the **human-robot co-worker** scenario, addressing the **human-robot interfacing and safety** research focus.

In this report we describe how to Install, configure and use the first Prototype V 1.0 Released in the mid-term milestone (Month 10).

1 Requirements

1.1 Speaky Platform Requirements:

O.S: Windows 7 Operating System - English

1. Install SpeechPlatformRuntime.msi (x86)
 2. Install MSSpeech_SR_en-US_TELE.msi
 3. Install MSSpeech_TTS_en-US_Helen.msi

LINK

<http://www.microsoft.com/en-us/download/details.aspx?id=27225>

<http://www.microsoft.com/en-us/download/details.aspx?id=27224>

4. Install vcredist_x86_2005.exe, vcredist_x86_2008.exe, vcredist_x86_2010.exe
 5. Install VisualBasicPowerPacksSetup.exe
 6. Install .NET Framework 4 (you can use Web Platform Installer)

LINKS:

http://w

1.3 Specialty Command Manner Requirements

In order to have the CommandMapper working properly, you must

of the Java JDK or JRE with the right environment variables set.

WWW file can be downloaded at <http://www.msw-pro.org>

Attention:

1. The version of SWI-Prolog and the Java JRE/JRE with respect to the architecture (32/64bit) MUST BE the same
 2. The file `sem_extractor.pl` and the folder `/frames` MUST BE in the same folder of the `SpeakyCommandMapper.jar`

1.3 Tools

The following tools useful tools for testing and debugging but are not mandatory so this section can be skipped.

1. 7-ZIP
 2. Notepad++ (npp.6.1.3.Installer.exe or above)
 3. DebugView (DebugView.zip). (unzip the file in your preferred directory i.e. Desktop)

LINKS:

<http://www.7-zip.org/download.html>

<http://notepad-plus-plus.org/download/v6.1.5.html>

<http://technet.microsoft.com/en-us/sysinternals/bb896647.aspx>

2 Speaky 4 Robots Installation

2.1 Speaky Platform Installation

The Speaky platform is an important part of the speaky for robots project and contains many customization of the original Mediavoice Speaky Platform specifically developed for this project so it has to be installed as a main component of the Speaky 4 Robots.

To install the Speaky platform and finalize the installation the following steps must be taken:

1. Run the following Command Line: "SpeakyPlatformSetup.exe D1DD668E-3C15-4b2a-95F6-696FC92967E9 AAAAAA-46834-84708-93829"
2. replace the SpeakyASRServer.exe overwrite using the one you'll find in the package.
3. save a copy of SpeakyTtsServer.exe to another folder
4. rename SpeakyTtsServerMicrosoft.exe as SpeakyTtsServer.exe, thus replacing the one at point 3.

Speaky Broker Installation:

1. Copy folder "SpeakyBroker_f68ed78c-09db-4a1f-95d1-f9e20707a83f" under C:\Users\USER_NAME\AppData\Roaming\Speaky\speaky_applications
2. Copy SpeakyBroker.exe under any folder i.e. C:\Users\USER_NAME\AppData\Roaming\SpeakyBroker

Configurations to finalize the Speaky Platform and Broker Installation:

1. open config.xml under C:\Users\USER_NAME\AppData\Roaming\Speaky

Set the following entries:

Under <section name="asr params">
 <entry name="lang">en-US</entry>

Under <section name="tts params">
 <entry name="voice">Helen</entry>
 <entry name="lang">English</entry>

Under <section name="sezione principale">
 <entry name="applicazione di default">f68ed78c-09db-4a1f-95d1-f9e20707a83f</entry>

Under <section name="asr params">

Add/Modify the following Entries:

```
<entry name="audio dump">false</entry>
<entry name="dump path">YOUR_PATH</entry>

<entry name="use_semantic_analyzer">true</entry>
<entry name="use_double_check">false</entry>

<entry name="semAnanyzerIpAddress">127.0.0.1</entry>
<entry name="semAnanyzerPort">27000</entry>
```

2. open the following xml file

C:\Users\USER_NAME\AppData\Roaming\Speaky\speaky_applications\SpeakyBroker_f68ed78c-09db-4a1f-95d1-f9e20707a83f\XML\SpeakyBroker_f68ed78c-09db-4a1f-95d1-f9e20707a83f.xml

set the entry <path>C:\prj\Demo Speaky Broker\Speaky Broker - VS2010\bin\Debug\SpeakyBroker.exe</path>
to point to your executable. see Speaky Broker Installation section above.

i.e. C:\Users\USER_NAME\AppData\Roaming\SpeakyBroker\SpeakyBroker.exe

3. copy the following files

1. grammar.grxml
2. grammar.cfg
3. Positional_description_syntSemTreeProlog_v1.1.grxml
4. Positional_description_syntSemTreeProlog_v1.1.cfg

to the following folder

C:\Users\USER_NAME\AppData\Roaming\Speaky\speaky_applications\SpeakyBroker_f68ed78c-09db-4a1f-95d1-f9e20707a83f\GRM\

2.2 **Command Mapper Installation (Semantic Analyzer)**

The CommandMapper is a Java .jar file that works as a server/client bridge, communicating with the Speaky ASR Server. Its purpose is to analyze the different outputs of the Speech Recognition process, and to return the most appropriate one, encoding it using a XML representation of the result.

Setting the Environment Variables

Furthermore, you need to set the following environment variable:

1. **SWI_HOME_DIR**: pointing to the SWI-Prolog installation folder
EXAMPLE: C:\Program Files\pl
2. **Path**: you have to add the path to the \bin folder of the prolog installation folder
EXAMPLE: C:\Program Files\pl\bin

3 Using the Prototype

In order to use the prototype you have to launch the following applications in the order as they are listed below:

1. SpeakyCommandMapper
2. Speaky Platform

3.1 How to launch Speaky and CommandMapper

1. Launch the SpeakyCommandMapper typing the following command:
 - a. C:\YOUR_PATH_TO_COMMANDMAPPER\java -jar SpeakyCommandMapper.jar
2. Speaky starts automatically when you turn on your computer. To start and stop Speaky, you can double click the Speaky Control Panel icon on the Tray bar. Once the Control Panel is opened, you will see the start and stop button that perform the previous tasks, located in the Speaky Tab Panel. When you stop Speaky, remember to close manually the Speaky Broker, because it will start automatically when you'll restart Speaky. This to avoid having two Speaky Broker processes running at the same time.

3.2 How to give voice command to the Speaky For Robots Prototype

Once Speaky has been launched, the “welcome” voice prompt is delivered. From now you can give voice commands to the robot.

To give a voice command just press and release the spacebar key and speak naturally one of the following voice command:

“Go to the kitchen”
“Move to the hall”
“Walk near the window of the sitting room”
“Run to the television”
“Move near the chair on your right”
“Go the dining room”

3.3 Technical Support

For any problem on the installation or use of the prototype please contact manes@mediavoice.it

Task 3: Robotic Voice Development Kit Design [M 6-12]

Participant	Role	Person-months
MV	Leader	14
UR1	Participant	8

Objectives:

This task describes the activities necessary for the development of the Robotic Voice Development Tool (starting from design process):

- Implementation and testing of the communication protocol (Integration System) between Speaky and the Robot environment, to transmit the voice commands received by the user to the robot, and, conversely, to transmit the results of operations performed by the robot to the voice interface developer (by the TTS).
- Implementation and testing of the wizard, which will guide the voice interface developer in the definition of the VUI, and then, must produce the data structures for the TTS and for ASR, called: Grammars, set of allowed words; Vocal Prompts, set of sentences to be read after an action; Vocal States, set of states that specify which grammars should be active and when.
- Implementation and test of the voice interface generator, which builds the input for the speech processing system based on the input acquired through the wizard.
- Testing of the interface generation tool by applying it to the two case studies in home service and outdoor robotic surveillance.

The output of this task will support the activities conducted in T4.

Description of work, and role of participants:

T3.1: Infrastructure System Development

In this first part of the task the participants, MV and UR1, integrate the different components of the system. The work will be split up into different stages:

- Development and implementation of the communication methods (physical and logical protocols) and of the selected formalisms (descriptive language and related features), which the subsystem exchange information with, using dimensional parameters coming out of the previous phase of the analysis.
- Development of the physical Database, repository of the data shared between the system components.

T3.2: Wizard Development

MV will mainly carry out this part. The work will be split up into different stages:

- Development of the voice templates, related to the grammars, the voice prompts and the vocal states, that will be used and filled by the wizard, depending on the voice interface developer's needs;
- Implementation of the wizard in accordance with the designed diagrams resulting from the analysis, so that as the user chooses his preferences, the system fills the templates necessary for the voice interface and gives the reader instructions to select the data necessary for the following choices;
- Consistency/operational test and wizard calibration;

T3.3: Voice Interface Development

MV will mainly carry out this part.

The work will be split up into different stages:

- Analysis of the designed diagrams and selection of the best environment for the implementation, taking into consideration the formalisms used in the ontology representation.
- Implementation of the design produced by T2 related to all the generation steps with a careful analysis of the output (relationship between linguistic specification and voice commands);

T3.4: Voice Interface Testing

UR1 will mainly carry out this part. The work will be split up according to the two different case studies in home service and outdoor robotic surveillance.