European Clearing House
for Open Robotics Development
www.echord.info

ECHORD

Technical Report D1.1:

Robotic Domain Definition and Representation

E. Bastianelli*, G. Castellucci*, F. Giacomelli**, N. M. Manes**, L. Iocchi*, 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 the results of the activity carried out in Task 1 (description reported in the Appendix).

1 Introduction

In this report we describe the results of the first activities of the project that are focussed on the analysis of the proposed application: a Robot Voice Development Kit, aiming at enriching the Speaky Development Environment from Mediavoice with a set of functionalities for the implementation of voice user interfaces that are customized for specific robotic platforms and specific application domains. The vocal interface that is targeted by S4R relies on the use of a generic ASR, namely on a speech processing engine. Specifically, S4R is developed in order to enrich the development environment Speaky, which is currently built around the Loquendo ASR. Speaky is designed to support other speech engines, as well as, networked speech services.

The analysis has addressed all the aspects that in the proposal were identified as components of the proposed solution: (1) the definition of a specialized vocabulary and its dependencies from the platform and from the application domain; (2) the use of domain knowledge to broaden the language understanding capabilities; (3) the exploitation of the ASR functionalities to improve the performance of the speech understanding process.

The following sections of the document are organized according to the above outlined structure, while in the concluding section, we sketch the architecture of the proposed solution to be fully exploited by the design activities that are planned within Task 2 and summarize the outputs of this phase of the project.

In the rest of this section, in order to provide an up-to-date context for the project development, we present an updated discussion of the state of the art, specifically analysing recent developments in the use of voice technologies for Human Robot Interaction, which has received a significant attention in the last year (see for example [Issue, 2011]).

We structure our analysis in two parts, the first one addressing the systems that provide vocal interaction on robotic platforms and then we focus on systems and proposals that consider Natural Language Processing Techniques in the context of robotic applications.

1.1 Voice user interface for robotic systems

The approaches aiming at the use of vocal interaction with robots considering both commercial robotic platforms and research prototypes fall into 3 categories:

- Dedicated speech processing through ad hoc proprietary systems for vocal interaction
- Use of commercial off-the-shelf tools for Automatic Speech Recognition (ASR)
- Use of networked services for speech recognition

In the first category, we have for example robuBOX-Kompaï by Robosoft [Sallé et al., 2007], for elderly people assistance, that provides basic speech recognition and synthesis, for simple tasks. In particular, we have seen developments on the NAO Humanoid platform by Aldebaran Robotics [Aldebaran], whose latest versions include both a speech recognizer and an application for determining the direction of the speaker.

Dedicated robotic speech systems are hard to evaluate, since they typically require the availability of the robotic platform. They undoubtedly suffer from the limitation in the computational power of the hardware on board of the robot, but on the other hand they may take advantage of the knowledge about the robot capabilities in order to focus the interpretation of voice commands and the interaction with the robot, for example by integrating semantic disambiguation in the speech understanding process (see for example [Wang et al., 2005]). It is also unclear to what extent they support a modular design that allows for reuse, adaptation, customization on different robotic platforms and different application domains. Since the NAO platform is adopted as one of the robotic platforms used in the experiment, it may provide a reference for comparison (the currently available NAO platform does not provide the functionalities for vocal interaction). In addition, many research approaches on voice interaction with robots rely on dedicated speech systems (see for example [Issue, 2011] and [Scheutz et al., 2011] therein).

Nowadays, several approaches to Automated Speech Recognition (ASR) in robotics are achieved through the use of generic tools for speech processing (e.g. CMU Sphinx, IBM ViaVoice). Commercial off-the-shelf Automated Speech Recognition systems explicitly designed for robots are still not available. An example of robot, using an ASR system, is [PI Robot], is built on ROS and relies upon the CMU Sphinx toolkit. The Loquendo ASR, which is currently embedded in Speaky, is one of the general purpose speech recognition products. It is worth noticing that Speaky is designed in such a way that the internal speech engine can be replaced. It is foreseen that the solution developed by S4R will be tested also with a different speech engine.

The typical approaches to voice interfaces for robotic applications with off-the-shelf speech systems are usually simple, meaning that the language is rather limited and the tools are used as black boxes, with little understanding on how to fully exploit their capabilities in terms of the broadness of the language and of its performance (an exception is for example [Doostdar et al., 2009], where multiple decoders are used in order to improve performance).

The goal of S4R is to exploit the capabilities of the speech engine by both specializing the input specification of the language and to analyse the output of the speech processing, in order to improve the interpretation by relying on the semantics of the domain and on Natural Language Processing (NLP) techniques.

The third category of possible approaches to the design of a voice user interfaces is rather recent and relies on the new speech understanding services, typically offered for mobile devices. Since these services have been launched very recently, little experience is available; however, there are expectations that they will allow for a significant step forward in the use of voice user interfaces. These speech recognition services are supported by cloud computing, thus allowing for a powerful computation and enabling additional services for the end-user. Google Speech Recognition included in the Android SDK and Nuance Developer Network SDK are the most popular ones.

Among the issues that need to be addressed in order to verify the applicability of networked speech services to the design of voice user interface for robotic platforms are the performances in terms of the interpretation and the impact of possible delays and failures in communication. As a consequence of the availability of this additional class of solutions, it is foreseeable that within S4R, also embeddings with networked speech services will be tested. Mediavoice is in fact actively working on the exploitation of ASR capabilities on mobile phones, both in terms of local processing and of networked services. This also opens the possibility of adopting a mobile phone as the device for user interaction, since it can be used as a voice terminal, as well as a tangible user interface for controlling the robot [Randelli, 2011].

Recent research works that address voice used interfaces for robotic systems can be characterized by the type of proposed approach (grammar-based versus HMM) and for the purpose of the interaction (command vs dialogue for learning or other communicative action). A recent comprehensive set of reference is given in [Scheutz et al., 2011]. Examples of specialized control languages are a voice control for wheelchairs [Nishimori et al., 2007], where the assisted person can command basic actions such as: moving forward, rotating, or stopping, and the more recent voice command language for a quadrotor [Huang et al., 2010] (see also [Tellex et al., 2011]).

1.2 Natural language processing for robotic applications

A significant body of work has been carried out in Natural language processing (NLP), whose intended application is a vocal interaction with robotic systems. Much of this work has been accomplished under the assumption that an ASR provides a textual input to be processed using NLP techniques. There is however, a substantial mismatch between the output produced by an ASR and the input considered by many proposals for NLP when starting from a textual specification. This notwithstanding, interaction with robots is the targeted domain for several applications of NLP and, in specialized contexts, and machine learning approaches have proven very effective.

NLP approaches targeting robotics applications have addressed a variety of issues. For example [Issue, 2011] includes papers about use of clues, turn taking, or reducing uncertainty through dialogue. Moreover, other work focuses on the use uncertain and fuzzy concepts [Jayasekara et al., 2009], on pragmatic interpretation

according to a contextual model of the speaker's behaviour [DeVault and Stone, 2009], on collaboration strategies between a naïve user and a robot [Foster et al., 2005], on integration with robotic behaviours [Brick et al., 2007], [Chen et al., 2009].

Below, we first specifically address the recent work on task oriented natural language interpretation, and then the work on the mapping of or referencing to elements in the environment.

In the first category, we find [Kollar et al., 2010] which presents an approach to build a robust natural language system aiming to understand direction commands. This process is realized by extracting shallow linguistic structure from the directions, grounding elements from that structure in the environment and performing inference to find the most probable path through the environment given the directions and observations. To model the linguistic input, the formalization of a structure is proposed by modeling each sentence in a set of direction as a hierarchy of Spatial Description Clauses (SDC). These clauses consist of a figure, a verb, a landmark and a spatial relation. In a sentence, verbs are the actions to take, landmarks represent objects in the enviroment and spatial relations are geometric relations between the landmark and the figure. The system takes the sequence of SDCs, a partial or complete semantic map of the environment (i.e. a map annotated with the locations of the objects used in specifying directions), a starting location and it outputs a sequence of waypoints through the environment, ending at the destination. The SDC is extracted from text using a Conditional Random Field (CRF), while planning is formulated as finding the maximum probability sequence of locations in map. This inference uses a probabilistic graphical model that is factored into three main components. The first component grounds novel noun phrases in the perceptual frame of the robot by exploiting object co-occurence statistics between unknown noun phrases and known perceptual feature. These statistics are learned from a large database of tagged images. The second component realizes a spatial reasoning that judges how well spatial relations describe a path. The last component models the verb phrases according to the amount of change in orientation in the path. In [Tellex et al., 2011] the above approach is extended to other types of commands, by introducing the concept of Generalized Grounding Graph (G^3) to deal with variable arguments or nested clauses in the command sentence. The structure of the graph model is induced using SDCs, representing a linguistic component of the command that can be grounded to the world (an object, place, path or event) from a semantic map of the environment. The model is trained on a corpus of natural language commands paired with groundings for each part of the command, enabling the system to automatically learn meanings for words in the corpus, including complex verbs. At the top level, the system infers a grounding corresponding to the entire command, which is then interpreted. This approach can be applied to domains, where linguistic constituents can be associated with specific action and environmental features.

In the second category address the use of natural language is targeted to enabling the user to reference elements in the environment, in robotics, often referred to as the process of “grounding” [Hertzberg and Saffiotti, 2008]. This is in particular used for example to build a semantic map of the environment or to specify commands.

[Kruijff et al., 2007] propose an architecture to enable robots to augment their autonomously acquired metric map with qualitative information about location and objects in the environment, through interactions with a human. For this purpose an ontology-based approach to multi-layered conceptual spatial mapping is presented, that provides a common ground for human-robot dialogue. A multi-layered representation of the environment is adopted, combining metric maps and topological graph, extended with conceptual descriptions that capture aspects of spatial and functional organization. The latter are obtained either through interaction with human, or through inference combining observations with ontological knowledge. From the string representation of the commands, generated by a speech recognition component, a CCG parser is used to analyze the utterances syntactically and a semantic representation in the form of Hybrid Logics Dependency Semantics is derived. The overall architecture bridges the gap between the rich semantic representation of the meaning expressed by verbal utterances and the robot's internal sensor-based world representation.

In [Kelleher and Kruijff, 2006] a computational approach to generation of spatial locative expressions is presented, aiming to face the issue of combinatorial explosion inherent in the construction of relational context models. This is achieved by defining the landmarks within the set of objects in the context and sequencing the order in which spatial relations are considered, using a cognitively motivated hierarchy of relations, and visual

and discourse salience. The goal is to develop conversational robots capable of interacting through natural and visually situated dialog.

[Ros et al., 2010] address the problem of grounding a referent (e.g. an object) with an efficient human-robot interaction. A robotic system should in fact be able to use some strategy to find a referent autonomously. The authors found two main reason for this: 1) humans are not always aware of when they introduce ambiguities; 2) if a robot is not able to ground a referent by itself, it has to inquire the human for clarification which would result in a tedious human-robot interaction. The paper describes an approach to taking a visual perspective (that is the ability of perceiving the environment from other's point of view), in order to find a referent based on a set of descriptors, when incomplete/ambiguous information has been provided by a human partner.

2 Vocabulary

ASRs typically acquire the input using a grammar specification in one of the standard specification languages. The language that is adopted by Speaky and, consequently we adopt in S4R, is the Speech Recognition Grammar Specifications 1.0 W3C Recommendation (SRGS) as defined in [W3C, 2004], which is based on an xml syntax. Moreover, the mechanism used to return the interpretation of the vocal input is based on attributed grammars: Loquendo ASR uses the Semantic Interpretation for Speech Recognition (SISR) W3C Candidate Recommendation [W3C, 2006], and, consequently, S4R does. The specification is relatively easy to provide, and we have decided to start acquiring some experience on the language definition, by implementing a simple prototype version of the voice interface for the Aldebaran NAO robot and the Videre design wheeled robot [1].

In the next subsection, we briefly describe these first prototype implementations. However, the language resulting from a straightforward specification blows up exponentially as soon as one attempts to provide some degree of flexibility to the language, and to broaden the vocabulary to accommodate a variety of environments, tasks and robotic platforms. Consequently, a rational design of the input specification for the ASR becomes the key technical and scientific issue that is targeted by S4R, in order to improve the performance of the speech understanding process. More specifically, our goal is to provide a modular definition of the language that makes it customizable for a variety of robotic platforms and applications. By relying on a modular definition, we aim at providing an input specification for the ASR that is specifically targeted to a platform/application deployment of the interface and, consequently, improve the performance of the speech recognition process. To this end, we have built two collections of sentences representing commands in the two application domains addressed by S4R and we have analysed them in order to provide a basis for structuring the modules. Moreover, we have analysed the state of the art linguistic resources and knowledge specifications available. In Subsection 2.2, we report on the analysis of the collected commands, while the use of the domain knowledge from external sources is addressed in Section 3.

2.1 First prototype speech interfaces

In order to fully explore the capabilities of the Speaky system and to ground our development onto the robotic platforms, we realized two prototype implementations of the vocal interface for the chosen platforms.

For the first prototype we used the NAO humanoid robot by Aldebaran Robotics and choose the robot-soccer as test domain. Using the robot-soccer domain, we were able to test the motion commands like, walk, turn left/right, stop, and the commands for moving the head like watch left/right, since they are readily available on the robot. Moreover, we also tested more complex commands that are needed to play soccer and require complex behaviors and, possibly also perception (i.e. kicking the ball). Needless to say, the basic commands are not very effective for controlling the soccer player, even if supported by “human” vision. This prototype has been used for a number of experiments to test the flexibility of the grammar definition (see Section 4).

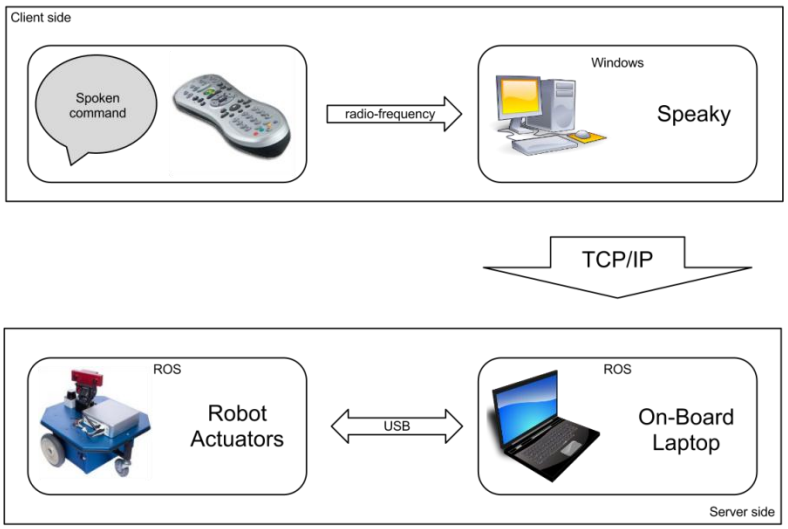
For the second prototype we used the Erratic wheeled robot by Videre Design that we have designed to test an indoor surveillance application. Being the Erratic a differential drive robot, its capabilities are limited to motion commands in 2D environment, again ranging from the basic, move forward/backward, turn left/right to the more

[1] Since the experiment started before the initial financial instalment, we are using the robotic platform designed al UNIRM1 for surveillance in indoor environments

usable “goto position”. Moreover, we have defined also commands to control the speed and commands to acquire information from the sensors. Despite the relative simplicity of the commands given to the robot, this prototype was also very useful to understand how to actually embed the speech interface on a more conventional software robotic architecture. In particular, it allowed to test the compatibility of our system with ROS, which is quickly becoming the standard operative system in the robotic field.

The implementations are basically composed by two elements: the first one includes the Mediavoice terminal, through which the audio is acquired, and a PC running Speaky and processing the audio input; the second one is the robotic platform with its on board computer. These two elements are connected via a client/server architecture on a TCP/IP connection. The PC running Speaky acts as a client sending commands to the robot as a result of the speech recognition and the robot acts as a server waiting for commands to be executed.

Through a TCP/IP connection, we are also able to replace the robot with a simulator. In particular, we used GAZEBO, a 3D simulator available under when using ROS. The use of a simulator can be very convenient also for the early stage of the sperimentation of the voice interface developed by S4R.



The above sketched architecture achieves a very effective decoupling between the implementation of the interface and the robotic platform. This facilitates the integration with existing robotic platforms, however, it requires an additional computing unit to handle the vocal input and its placement on board of the robot may not be feasible. A suitable hardware software configuration will be addressed in the subsequent stages of the project. The following picture shows the second prototype being tested in our laboratory [2].

[2] Videos of both prototypes are available at http://www.youtube.com/watch?v=WxiY5YlZOzA and http://www.youtube.com/watch?v=szdMPj_YbVk).

2.2 Modularizing the vocabulary

In order to structure the language specification in a modular way, we apply both a bottom-up approach, starting from examples of the commands taken from the users and a top-down approach, by exploiting linguistic and knowledge resources already available for the domains on interest. Below, we describe the proposed structure of the language devised by analysing two collections of sentenced representing commands that have been built by the students of the course Seminars in Artificial Intelligence and Robotics of the Laurea Magistrale (Master) in Artificial Intelligence and Robotics as a part of their NLP section requirements.

It is worth clarifying that our goal in collecting these sets of commands was NOT to provide a systematic basis for language elicitation; rather, our aim was to gather some general guidelines for structuring the knowledge. While some recent works reported in the literature (see for example [Tellex et al., 2011]) propose a systematic approach for learning the commands from a corpus, such an approach would require a substantial effort for each specific customization of the language, which is currently not in line with the industrial objectives of Mediavoice. Subsequent work in S4R on the evaluation of the proposed approach will address the trade-offs between a learning approach and top-down design of the application.

The first collection contains commands for a robot in a domestic environment. Although constrained, a home environment is a very challenging one for the capabilities of a robot (see for example [Tenorth, 2011] [Pronobis, 2011]). Intentionally, we did not provide a detailed specification of the capabilities of the robot and favoured a broad language definition. Thus, the collection contains also a number of commands that are not applicable to the robot, although relevant to the domain. This should eventually enable the interface to recognize the commands, and properly answer that a requested command is not executable. In order to enrich the collection of sentences and address additional aspects of the language for this domain, we plan to interview the teams of the RoboCup@Home competition, where a voice interface is one of the capabilities that are required for participating robots.

The second collection contains a set of commands for the interaction with a rescue robot. Again, we have focussed on a broad language, without restricting to the actions executable by a specific platform. The students who collected the commands had attended the first section of the seminar course on Rescue Robotics and were therefore familiar with the capabilities of rescue robots, as well as with the possible uses of a robot in a rescue operation.

As expected, the analysis of the collections of commands shows a large variability along several directions. Verbs are naturally related to the commands executable by the robot, and one can group them according to the hardware structure of the robotic platform (e.g. walking, grasping, pressing buttons) and its perceptive capabilities. From the point of view of the robot capabilities, there are also significant differences in terms of abstraction of the action (“walk forward”, as opposed to “go to the kitchen”). In fact, the collection contains several examples of sentences that require a sequence of actions, possibly combined with perception; often such actions are not directly available as commands in our reference platforms (but also in the typically available ones).

The other components of the sentence that are of interest typically provide information about the execution of the command and can be reflected into arguments of the command. In this respect, the semantics and structure associated with the verb appear to be very relevant (see next section), in order to extract other information needed for the execution of the command. However, it is worth emphasizing the large variety of wordings that can be used for example to denote the location to be reached by a motion command. Here a characterization of the environment seems to play an essential role and it is also relevant to distinguish those elements of the environment that can be handled by the robot for a proper assessment of the command language. To this end, it would be extremely valuable to rely upon a semantic map of the environment [Pronobis 2011], [Hertzberg and Saffiotti, 2008], [Randelli, 2011].

Besides the environment and the robotic platforms, also the specific task accomplished by the robot does provide additional linguistic elements. In a rescue scenario, the user can refer to “victims”, while in a surveillance scenario it maybe more likely to find elements such as “intruder”. Moreover, in a home the environment is assumed to be known and references to locations are usually direct (“check the entrance door”) as opposed to a rescue scenario, where the environment is unknown and typically described with respect to the

field of view of the robot (“enter the door on the right and take a snapshot”). Finally, since the rescue robots are tele-operated, the commands typically include several low-level controls, like the speed of the robot or the direction that the camera must point to. Summarizing, while there are several dependencies from the task that is addressed by the robot, taking them into account would require an additional analysis of both the robot capabilities and the features of the environment.

3 *Semantic* Representation

In this section we discuss the representation of the semantic knowledge associated with the platforms the domains and the tasks of each specific contextualization of a robotic application. There have been a number of proposals to define domain ontologies in the case of Service Robots (see for example [Tenorth, 2011] [Hertzberg and Saffiotti, 2008]). Moreover generalized knowledge resources such as CYC [Matuszek et al., 2006], contain knowledge about the home environment. Also in the case of rescue robots there have been attempts to characterize the robotic platforms, as well as the domain (see for example [Fanelli et al., 2006], [Chatterjee and Matsuno, 2005]). Moreover, NIST has developed a taxonomy of robotic platforms and sensors for Rescue Robotics, within their simulation environment USARSim [Balaguer et al., 2008]. A variety of representation languages are used in the above cited approaches, ranging from specialized languages for representing ontologies (i.e. OWL) to rule-based languages and there are no consolidated reference standards. Consequently, reuse of existing knowledge is impractical.

On the other hand, linguistic resources, such as FrameNet [FrameNet], which also embody a representation of the semantics of linguistic terms, are widely used. They also seem to suit S4R goal of exploiting general knowledge for broadening the possible wordings of the commands for the robots. Moreover, FrameNet has already been used in the RoboFrameNet project [Thomas and Jenkins, 2011], whose goals are closely related to S4R. In the following, we first provide a sketchy description of FrameNet and then address in more detail RoboFrameNet.

FrameNet is a project started at the Berkeley University [FrameNet] aiming at the development of a linguistic resource for the semantic annotation of English, based on the Frame Semantics theory [Fillmore,1985]. Its semantic formalism is based on a structural representation of a situation/event evoked by a word (or a group of words) in a sentence, the so-called lexical unit. Every frame provides a set of frame elements (representing roles), whose purpose is to represent additional aspects of the situation described. As an example, consider the sentence “go quickly to the kitchen”, in which a motion event is described. We can recognize two roles: the former is where to go (kitchen) and the latter is how to execute this movement (quickly). In Table 1 a representation of this sentence is reported. The roles defined for each frame are grouped in two main subsets: core roles and non-core roles. The former are the most critical for the understanding of a sentence.

FrameNet defines also a very large corpus of English annotated sentences (about 175.000 sentences). The corpus enables the construction of a ML approach to the so called *Semantic Role Labeling* task, that is the process of extracting the semantic structure expressed within a sentence, in this case according to the FrameNet formalism.

Sentence	go to the kitchen quickly	
Lexical unit	go.v	
Frame	Motion	
Roles	Goal	kitchen
	Duration	quickly

FrameNet formalism can be used to represent the semantics of commands expressed in natural language to a robot as shown in Table 1. Frames can be extracted from descriptions of the capabilities of a robot and from an analysis of the possible dialogues that are likely to happen between a human and the robot. New frames can be defined and existing frames can be adapted to match the structure of robot commands. Frames can be used in order to generate the grammar specification for the ASR, as well as to interpret or disambiguate the output of the ASR. Partial matching with the frame structure can also provide useful goals for the dialogue with the robot.

The above sketched approach to semantic representation of robot commands has been proposed in the RoboFrameNet project [Thomas and Jenkins, 2011], based on the idea of using semantic frames. Specifically, the goal of RoboFrameNet is to obtain an abstract frame from a sentence in natural language and then to ground it into goal oriented robot actions. In order to abstract from a sentence to a frame, the audio input is converted to a text using a networked service for voice recognition. Once a text is available to the system a semantic tree of the sentence is generated using the Stanford parser [Stanford]. The last step of the abstraction is made by the so called "Semantic Framer", which maps the information of the semantic tree to the corresponding frame. In order to obtain the desired mapping, RoboFrameNet makes use of lexical units that are hand-annotated mappings of each word in a sentence to relevant elements in a frame. Ultimately, these lexical units will be collected in a corpus from which humans and machines can learn to create their own mappings.

RoboFrameNet indeed pursues an approach that matches very closely the use of the linguistic knowledge that can be adopted by S4R. However, for speech recognition they rely on a networked service whose textual output is processed by a semantic parser in order to be correctly mapped to a frame. On the other hand, S4R relies on a generic ASR and focuses on the ability to customize the language specification for the ASR based on a semantic characterization of the domain knowledge. Indeed, a frame can be used not only to represent a command and ground it to an action as in RoboFrameNet, but also to enrich the input grammar of the ASR. A simple way to achieve this is to start from a simple representation of an action, for example a verb, and look for all the possible frames related to it. Then, from the frames retrieved and their roles, grammar rules can be derived in order to recognize the possible commands related to each of those frames.

This process is sketched below. Starting from a command the robot is able to execute, like "goto (position, speed)", a system can look into the knowledge base for the command representation and find the association with motion frame for the command goto. The next step is to look up into the collection of frames derived by RoboFrameNet and check for the ones representing the idea of motion. In this case, we will find three of them namely moving and its two children moving_backward and moving_forward. The description of frames in RoboFrameNet is currently somewhat ad hoc, and FrameNet can be used instead. Its frame related to motion, is the following:

```
Frame Motion
 Lexical Unit: go.v, move.v, slide.v, drop.v, rise.v
 Core roles:
   Goal: the location the Theme ends up
   Path: the Path refers to the ground over which the Theme
Travels or to a landmark by which the Theme travels.
   Source: the location from which the Theme moves
   Theme: the entity that changes location
takes Place
   Manner: identifies the Manner in which the Motion takes place
Time: identifies the Time when the Motion occours.
```

This structure provides a bridge between the command and its linguistic counterparts. Moreover, the process can be refined by considering other verbs that are related to the motion frame, e.g. move, slide. The distinction between core and non core roles can be instrumental to characterize partially specified frames.

Moreover, frames can also be used in order to disambiguate between partially recognized sentences. When the whole sentence is not recognized, but it is still possible to associate it to a frame, one can use the missing roles of that frame to ask for more detailed information. More specifically, S4R aims at supporting also the subsequent dialogue with the user, by exploiting the frame structure. For example, if the ASR recognizes the sentence "go quickly", by referring to the lexical unit (go), the partial matching with the corresponding motion frame suggests to acquire the missing piece of information. If the goal location can not be recovered from the state of the dialogue, it can be asked to the user.

Finally, RoboFrameNet is an open project under development and there is a possible synergy with S4R.

4 Grammar-based Specification

In this section, we address the results obtained by providing the specification of the language to be handled by the ASR engine through a grammar-based specification.

The prototypes illustrated in Section 2 have been tested, first with the ASR in Italian, and subsequently with the English ASR. In particular, we have tested a grammar for NAO robots playing soccer and we have acquired some evidence that, when the grammar is designed to allow some degrees of flexibility, for example arbitrary ordering of the main components of the sentence (i.e. “strong kick left” vs “kick strong left”), the performance of the speech interpretation process is substantially impacted.

Based on this first experience, we have analysed the capabilities of Speaky, trying to identify possible ways to improve the recognition process, when the language is broadened. In addition, we have identified the need for a tool supporting the analysis of the input language and of a tool supporting the systematic evaluation of the system performance.

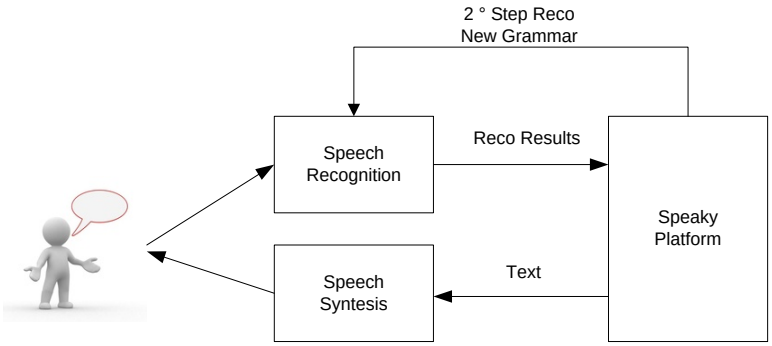
With respect to the first issue, in order to define the input specification one should take into account some generic linguistic components that are typical of the interaction, such as for example “Please” in “Please, go to the kitchen”. These components are dealt with in Speaky and are included in the grammars definition process; we do not further investigate them in the current project.

In addition, we have considered the following extensions of the Speaky development environment with the aim of improving the recognition process.

Dynamic grammar definition: the grammar used by the ASR need not to be statically defined, but it can be build dynamically, namely specified based on the current state of the system. This capability, which is already present in Speaky can be used in order to reduce the size of the grammar and thus allow for a more effective recognition. In previous applications developed with Speaky, this has lead to a pre-defined organization of the grammars that are loaded according to the state of the dialogue; in S4R, we can exploit this capability by taking into account also of the state of the robot within the existing dialogue manager.

Multiple result analysis: the ASR can provide in output different possible interpretations of the input speech, each with a confidence value. We plan to exploit this functionality by trying different approaches, to disambiguate the results produced by the ASR. To this end Speaky has been extended as to provide the results of the ASR in a format suitable for subsequent processing.

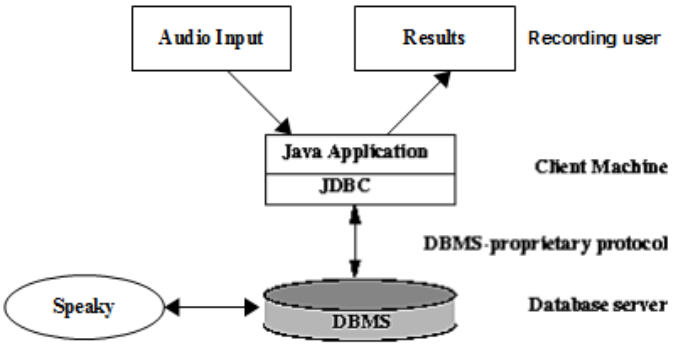
Repeated input analysis: since the result of the speech recognition is heavily dependent on the grammar used for the recognition process, by relying on the above illustrated dynamic grammar capability, we have considered the possibility of re-analyzing the input with a different grammar to verify whether a two step process can lead to an improvement of the performance in the recognition.



As illustrated in the above schema, Speaky interacts with the ASR, by providing the grammar dynamically, by processing the output of the ASR and, possibly, by requiring a second attempt to recognize the input on a different grammar.

With respect to the second issue, we are developing a suitable interface to obtain a confusability analysis, which allows for a preliminary evaluation of the grammar. Confusability identifies phrases in a grammar that may generate false acceptances on recognition, because of similarity to other phrases in the grammar. We are using several commercial tools to integrate into Speaky the confusability analysis [Microsoft Speech]. This task will give the designer some important feedback on: Which phrases are phonetically similar, Which phrases are likely to result in false acceptances, and What is the probability that one phrase will be falsely accepted for other phrases in the grammar. Confusability tells the developer whether the recognizer cannot reliably distinguish between multiple in-grammar phrases with distinct semantic meanings. Each result identifies a phrase that may be incorrectly recognized (and its semantics), the phrase with which it may be confused (and its semantics), and provides a metric for how likely it is that the confusion will occur. When calculating the probability that a phrase will be incorrectly recognized, the tool takes into account the weight assigned to each phrase.

With regard to the testing of the system, we have implemented a tool for managing input data (vocal commands by several speakers) and storing the results of the interpretation process. In this way, we can compare the performance of the system on the baseline specification, which is simply a list of commands, with the results of the system obtained after the customization based on domain knowledge. The architecture of the tool is illustrated in the figure below.



The collections of sentences can be pre-loaded in the system; the user, after identifying itself, can select different operation modes: either he/she is prompted a sentence to be vocalized, or he/she can select a sentence from the collection, or he/she can add a sentence to the collection. The vocal input is then both stored in the database and sent to Speaky for speech processing. The results of the ASR are visualized to the user and, optionally, stored in the database. Once the vocal inputs for a collection of sentences are recorded in the system, it is possible to re-analyse them and collect the results. The results of the processing can then be extracted from the database and several performance measures can be easily computed.

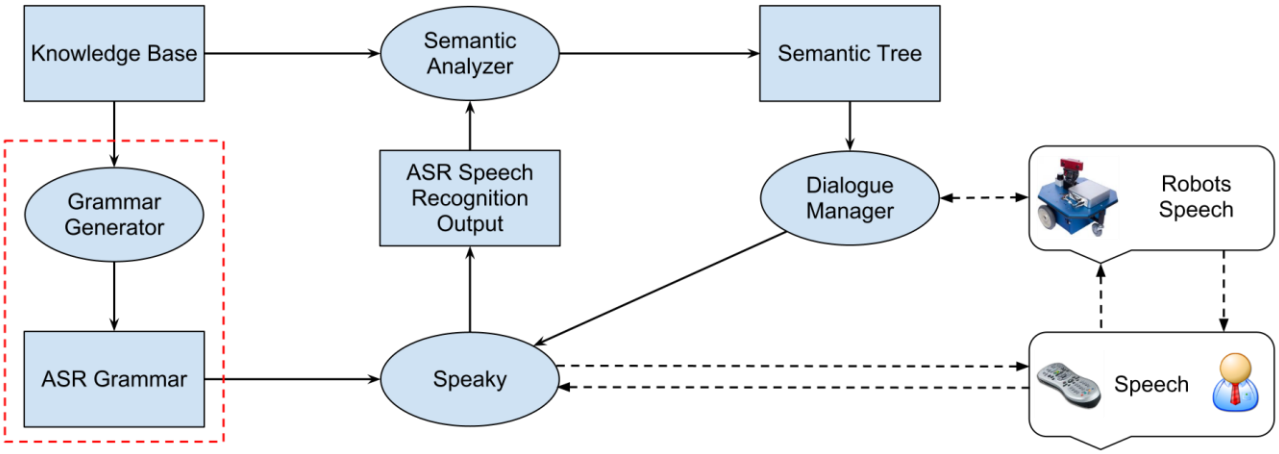
5 Proposed Architecture

In this section we describe the main components of the proposed solution. The rectangles in the diagram denote data/knowledge structures, while ovals denote the processes that manipulate them. The grammar generation process is accomplished off-line for each specific platform and deployment. The other processes become part of the run-time operation of the system. It is also worth recalling that the input is acquired by Speaky through a dedicated terminal, while the robot can produce speech either through an embedded Text To Speech (TTS) or through the TTS included in Speaky (part of the Loquendo toolkit). These flows of information are represented by dashed arrows between the user and the system (Robot/Speaky).

Below we describe the blocks in the figure in some detail. We start by describing the structures.

Knowledge Base: our approach aims at exploiting a priori knowledge about the robotic platform and the application domain. The knowledge is expressed in the ontology representation language OWL [OWL]. The

knowledge about the robot is structured following the robot ontologies available and characterizes the capabilities of the robot in terms of motion and sensing, including all the complex actions/behaviours that can be executed by the robot. This knowledge can also be extended with plans corresponding to the execution of more than one basic behaviour. In this respect, we do not explicitly include a planning component, but simply rely on libraries of plans. Similarly, the knowledge about the perceptive capabilities of the robot, should include a specification of the objects that can be perceived by the robot.



The knowledge about the application domain includes both knowledge about the environment and about the task to be accomplished by the robot. For example, the home domain would include knowledge about the rooms, their funtion, the objects that belong to them etc.. Indeed, the representation should be a suitable compromise between the breadth of language and the objects that can be actually be relevant to the robot capabilities. As for the task specification, if the robot is dedicated to a specific task, such as, exploration or surveillance, the description consists of a set of concepts representing situations and events that are relevant to the task.

ASR grammar: the input grammar for the Speaky recognizer is specified according to the specifications mentioned in Section 2, namely the Speech Recognition Grammar Specifications (SRGS) as defined in [W3C, 2004] and the Semantic Interpretation for Speech Recognition (SISR) [W3C, 2006].

ASR Speech Recognition output: the output of the ASR is a set of texts, each with an associated confidence value.

Semantic Tree: the semantic tree is a data structure characterizing the result of the interpretation process. It represents a partial or full command for the robot, if the overall process was successful. It is used by the dialogue manager, which is responsible of directing the actions of the robot, and/or additional interaction with the user.

Below we provide a specification of the processes.

Grammar Generator: the grammar generator is the process that, given in input the knowledge base and a specification of the chosen platform, outputs a grammar, more precisely a set of grammars to be dynamically used by speaky. This module supports the process described in Section 3. At this stage we are open to consider an interactive process, as well as the possibility of a fully automated one.

Speaky: this is the platform by Mediavoice, based on the Loquendo speech engine. It takes as input the grammar produced by the grammar generator and the vocal command of the user. As specified in Section 4, Speaky has already been extended with a number of functionalities to manage dynamic grammars, to produce in output multiple interpretations and to re-examine the input with a different grammar, and a module that handles the overall recognition process.

Semantic Analyzer: this module takes as input the Speaky's output and the knowledge base and builds the best inperpretation of the spoken command and its grounding to a robot executable command. In particular, this module tries to disambiguate on semantic grounds the output of the speech recognizer, when needed, and then it builds the mapping (total or partial) with a robot command.

Dialogue Manager: this module is already part of the Speaky run time environment, but we have outlined it explicitely, because it is responsible for the interaction with the robot. It sends the commands to be executed and it acquires from the robot information about the internal status (for example the command under execution), any alarm or perception that may require the attention of the user. As already mentioned, this information is used to determine the state of the dialogue and needs to be integrated in the existing Speaky dialogue system.

The detailed design of the above described architecture is the main activity for the completion of Task 2, which is expected to provide a the design of the Robot Voice Development Kit.

In conclusion, the first activities reported by S4R include:

- Two prototype implementations of a spoken command language, one based on the Erratic Robot, a wheeled platform developed by Videre, and one based on the humanoid robot by Aldebaran.
- The analysis of two collections of commands, related to the home and rescue test-domains. The collections, suitably enriched, will be used to test the capabilities of the system during its development.
- Several functionalities extending Speaky: a first group to handle the connection with the robots; a second group for improving the connection with the ASR; and a third one, to be used with the tool for preliminary analysis of the input specifications for the ASR.
- A tool to support a systematic performance evaluation of the proposed approach.
- An overall specification of the architecture of the Robot Voice Development Kit and its component modules.

References

[Aldebaran] Aldebaran. Robotics. URL http://www.aldebaran-robotics.com/ .

[Balaguer et al., 2008] Benjamin Balaguer, Stephen Balakirsky, and Stefano Carpin. Usarsim: a validated simulator for research in robotics and automation. *IEEE/RSJ IROS 2008 Workshop on "Robot simulators: available software, scientific applications and future trends"*, 2008.

[Brick et al., 2007] Timothy Brick, Paul Schermerhorn, and Matthias Scheutz. Speech and action: Integration of action and language for mobile robots. In *Intelligent Robots and Systems, 2007. IROS 2007.* IEEE/RSJ International Conference on, pages 1423-1428. IEEE, 2007.

[Chatterjee and Matsuno, 2005] R. Chatterjee and F. Matsuno. Robot description ontology and disaster scene description ontology: analysis of necessity and scope in rescue infrastructure context. *Advanced Robotics*, Vol. 19(8), pages 839-859, October 2005.

[Chen et al., 2009] X. Chen, J. Jiang, J. Guoqiang Jin, F. Wang, Integrating NLP with Reasoning about Actions for Autonomous Agents Communicating with Humans, in: *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 137-140, 2009.

[DeVault and Stone, 2009] David DeVault and Matthew Stone. Learning to interpret utterances using dialogue history. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 184-192, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[Doostdar et al., 2009] Masrur Doostdar, Stefan Schiffer, and Gerhard Lakemeyer. A robust speech recognition system for service-robotics applications. In *RoboCup 2008: Robot Soccer World Cup XII*, pages 1-12, 2009.

[Fanelli et al., 2006] L. Fanelli, A. Farinelli, L. Iocchi, D. Nardi, and G. P. Settembre. Ontology-based coalition formation in heterogeneous mrs. In *Proceedings of International Symposium on Practical Cognitive Agents and Robots*, pages 105-116, Perth, Australia, 2006.

[Fillmore, 1985] Charles J. Fillmore. Frames and the semantics of understanding. In *Quaderni di semantica*, Vol. 6(2), pages 222-254, December 1985.

[Foster et al., 2005] Mary Ellen Foster, Manuel Giuliani, Amy Isard, Colin Matheson, Jon Oberlander, and Alois Knoll. Evaluating description and reference strategies in a cooperative human-robot dialogue system. In *Proceedings of the 21st international jont conference on Artifical intelligence,* IJCAI'09, pages 1818-1823, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[Framenet] Framenet. URL https://framenet.icsi.berkeley.edu/fndrupal/home .

[Hertzberg and Saffiotti, 2008] Joachim Hertzberg and Alessandro Saffiotti. Using semantic knowledge in robotics. In *Robotics and Autonomous Systems*, Vol. 56(11) pages 875-877, 2008.

[Huang et al., 2010] Albert S. Huang, Stefanie Tellex, Abraham Bachrach, Thomas Kollar, Deb Roy, and Nicholas Roy. Natural language command of an autonomous micro-air vehicle. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2663-2669, Taipei, Taiwan, Oct. 2010.

[Issue, 2011] Special Issue. *Dialogue with Robots*, volume 34(2). 2011.

[Jayasekara et al., 2009] A.G.B.P. Jayasekara, K. Watanabe, K. Kiguchi, K. Izumi, Adaptation of Robot Behaviors Toward User Perception on Fuzzy Linguistic Information by Fuzzy Voice Feedback. In: *Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 395-400, 2009.

[Kelleher and Kruijff, 2006] J.D. Kelleher and G.J. Kruijff. Incremental generation of spatial referring expressions in situated dialogue. In *Proceedings of Coling-ACL '06, Sydney Australia*, 2006.

[Kollar et al., 2010] Thomas Kollar, Stefanie Tellex, D. Roy, and Nicholas Roy. Toward understanding natural language directions. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pages 259-266. IEEE, 2010.

[Kruijff et al., 2007] G.J. Kruijff, H. Zender, P. Jensfelt, and H.I. Christensen. Situated dialogue and spatial organization: What, where... and why? *International Journal of Advanced Robotic Systems*, Vol. 4(1), pages 125-238, 2007.

[Kruijff et al., 2006] G.J. Kruijff and Hendrik Zender. Clarification dialogues in human augmented mapping. In *Proc. of the 1st Annual Conference on Human-Robot Interaction (HRI06)*, pages 282-289, 2006.

[Matuszek et al., 2006] C. Matuszek, J. Cabral, M. Witbrock, and J. DeOliveira. An introduction to the syntax and content of cyc. In *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, 2006.

[Microsoft Speech] Microsoft Speech Platform Tools: http://msdn.microsoft.com/en-us/library/hh378385.aspx

[Nishimori et al., 2007] M. Nishimori, T. Saitoh, R. Konishi, Voice controlled intelligent wheelchair. In *Proceedings of SICE Annual Conference*, pages 336-340, IEEE, 2007.

[OWL] OWL Web Ontology Language Reference W3C Recommendation, February 2004 URL http://www.w3.org/TR/owl-ref/ .

[PI Robot] Pi Robot. URL http://www.pirobot.org/

[Pronobis, 2011] Andrzej Pronobis. *Semantic Mapping with Mobile Robots*. PhD thesis, School of Computer Science and Communication, KTH Royal Institute of Technology, Stockholm, 2011.

[Randelli, 2011] G. Randelli. *Improving Human-Robot Awareness through Semantic-driven Tangible Interaction*. PhD thesis, Università La Sapienza Roma, Italy, 2011.

[Ros et al., 2010] Raquel Ros, Severin Lemaignan, E. Akin Sisbot, Rachid Alami, Jasmin Steinwender, Katharina, and Warneken. Which one? grounding the referent based on efficient human-robot interaction. In *IEEE International Symposium in Robot and Human Interactive Communication (RO-MAN)*, pages pages 570-575, 2010.

[Sallé et al., 2007] D. Sallé, M. Traonmilin, J. Canou, and V. Dupourqué, Using Microsoft Robotics Studio for the design of generic robotics controllers: the robuBOX software, in: *ICRA 2007 Workshop Software Development and Integration in Robotics "Understanding Robot Software Architectures",* Roma, Italy, April 2007.

[Scheutz et al., 2011] Matthias Scheutz, Rehj Cantrell, and Paul W. Schermerhorn. Toward humanlike task-based dialogue processing for human robot interaction. *AI Magazine*, Vol. 32(4), pages 77-84, 2011.

[Stanford] Stanford Parser. URL http://nlp.stanford.edu/software/lex-parser.shtml .

[Tellex et al., 2011] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings Of The National Conference On Artificial Intelligence*, (AAAI), 2011.

[Tenorth, 2011] Moritz Tenorth. *Knowledge Processing for Autonomous Robots*. PhD thesis, Technische Universitat Munchen, Germany, 2011.

[Thomas and Jenkins, 2011] Brian Thomas and Odest Chadwicke Jenkins. Verb semantics for robot dialog.In *Robotics: Science and Systems Workshop on Grounding Human-Robot Dialog for Spatial Tasks*, Los Angeles, CA, USA, June 2011.

[Wang et al. 2005] Y.Y. Wang, Li Deng, and Alex Acero. “Spoken language understanding”. *Signal Processing Magazine*, IEEE, Vol(5). 22, pages 16-31, 2005.

[W3C, 2004] W3C. Speech recognition grammar specifcation version 1.0 w3c recommendation, March 2004. URL http://www.w3.org/TR/2004/REC-speech-grammar-20040316/ .

[W3C, 2006] W3C. Semantic interpretation for speech recognition (sisr) version 1.0 candidate recommendation, January 2006. URL

http://www.w3.org/TR/2006/CR-semantic-interpretation-20060111/ .

Task 1: Robotic Domain Definition and Representation [M 1-4]

Participant	Role	Person-months
UR1	Leader	6
MV	Participant	2

Objectives:

This task addresses the description of the considered robotic aspects, and the choice of effective representation formalisms, according to the following objectives:

- Definition of a **vocabulary** to provide common robotic terminology concerning: *(i)* robotic platforms, sensors and effectors, *(ii)* environments and *(iii)* applications. The vocabulary will be organized in core module and specific vocabularies. They will be defined for the two application scenarios considered in the experimental evaluation, including specific terminology about the possible tasks in such scenarios. This modular approach allows for **extending** the system, because any robotic expert can add modules, which will be then enrich the RVDK.

- S4R will investigate effective representation formalisms for semantic knowledge in connection with linguistic structures. In particular, the project will focus on lexical databases (e.g. FrameNet), where terms are arranged in **synsets**, and explicit semantic representations, such as **ontologies**. The aim is structuring information to exploit lexical and semantic term relations to provide enhanced functionality. Again knowledge will be arranged modules.

- The expressiveness of the speech-based robotic systems will be boosted up by adopting **syntactical grammatical constructs** to model complex robotic patterns. The goal is to provide a natural flexibility in the formulation of the requests of the user, while managing the complexity of the set of possible utterances that would reduce the performance of the speech recognition module.

The output of this task will support the activities conducted in **T2**. In fact, the whole system will be designed according to the adopted knowledge representation formalism, lexical vocabulary, grammar expressiveness, and syntactic structures. On the other hand, it will be independent with respect to the considered application scenarios.

Description of work, and role of participants:

T1.1: Application Field Vocabulary Definition
UR1 will evince lexical vocabularies from the case study of the two experimental scenarios defined in T4, thus adopting a bottom-up approach. This will be accomplished through interviews with participant users, to evince desired user requirements and typical intra-scenario activities.

T1.2: Core Vocabulary Definition
UR1 will define a core lexical vocabulary, by extracting common robotic aspects out of the considered experimental scenarios, such as: platforms, environments, sensors and effectors, behaviours.

T1.3: Formal Representation Analysis and Elicitation
UR1 will analyse the state of the art in terms of knowledge formalisms and will select an effective vocabulary representation, according to the desired user requirements and to the expected system performance. MV will collaborate at this activity evaluating the proposed formalism with respect to its integration ease with Speaky.

T1.4: Grammar Expressiveness Analysis
MV and UR1 will collaborate in evincing suitable grammars and basic natural language processing solutions. On the one hand, UR1 will seek an effective solution according to robotic requirements. On the other hand, MV will assess the integration feasibility of the proposed solution with the existent speech technologies.