Enhancing Safe Human-Robot Collaboration through Natural Multimodal Communication

Iñaki Maurtua, Izaskun Fernandez, Johan Kildal, Loreto Susperregi, Alberto Tellaeche, Aitor Ibarguren
Email: {inaki.maurtua, izaskun.fernandez, johan.kildal, loreto.susperregi, alberto.tellaeche, aitor.ibarguren}@tekniker.es
IK4-TEKNIKER
Parke Teknologikoa
C/ Iñaki Goenaga, 5 - 20600 Eibar
Basque Country, Spain

***Abstract*—This paper presents a semantic multimodal interaction approach between humans and industrial robots enhancing the dependability of the collaboration in real industrial scenarios. This is a generic approach and it can be applied to different industrial scenarios. We explain in detail how to apply it in a specific example scenario and how the semantic technologies help with accurate natural request interpretation leading to a more efficient collaboration, as well as its benefits in terms of system maintenance and scalability.**

I. Introduction

In modern industrial robotics, the safe and flexible co-operation between robots and human operators can be a new way to achieve better productivity when performing complex activities. Introducing robots within real industrial settings makes the interaction between humans and robots gain further relevance. The problem of robots performing tasks in collaboration with humans poses three main challenges: robots must be able to perform tasks in complex, unstructured environments, and at the same time they must be able to interact naturally with the humans they are collaborating with, always guaranteeing the safety of the worker.

Let us imagine an industrial collaborative robot and an operator in a deburring collaborative process. While the robot is deburring a piece, the operator has started with another one. The robot finishes the piece, and since the end of the operator working day is close, he decides to ask the robot to finish the piece he has started but not finished yet, while he finishes another tasks. There are several ways to rise such a petition. Following we list some of them:

- Manipulating the robot manually, positioning it just in the position the operator wants it to start from in the deburring task.
- Pointing at the area of the piece the operator wants the robot to deburr from, assuming the robot knows which operation it has to perform.
- Raising the petition via voice, indicating the task and somehow the position to apply it from.
- Combining voice and gesture, the first for indicating the action (i.e *Remove the burrs from this area*) and the latter for determining the area

978-1-5090-1314-2/16/$31.00

All of these communication possibilities are some of the research areas the *H2020 FourByThree*[1] project focuses on. The project aims at developing a new generation of modular industrial robotic solutions that are suitable for efficient task execution in collaboration with humans in a safe way, and are easy to use and program by the factory worker. The project will allow system integrators and end-users to develop their own custom robot that best answers to their needs. To achieve this, the project will provide a set of hardware and software components, ranging from low level control to interaction modules. The results will be validated in 4 industrial settings: Investment Casting, Aeronautical sector, Machining and Metallic Part Manufacturing, in which relevant applications will be implemented: assembly, deburring, riveting and machine tending in a collaborative context.

The present work describes the natural communication approach within the *FourByThree* European project. A requirement for natural human-robot collaboration including interaction is to endow the robot with the capability to capture, process and understand accurately and robustly requests from a person. Thus, a primary goal for this research is to analyze the natural ways in which a person can interact and communicate with a robot and go towards a natural, robust and reliable communication framework.

Natural communication between humans and robots can happen through several channels, the main of which are voice and gestures. In this multimodal scenario, the information can be complementary between channels, but also redundant. However, redundancy can be beneficial [2] in real industrial scenarios where noise and low lighting conditions are usual environmental challenges that make it difficult for voice and visual signals to be captured with clarity.

In this paper, we present a semantic approach that supports multimodal interaction between humans and industrial robots in real industrial settings that are being studied within the *FourByThree* European project. As mentioned earlier, the approach that we present is generic in the sense that it can be applied to different industrial scenarios by modifying the information about the environment in which communication takes place.

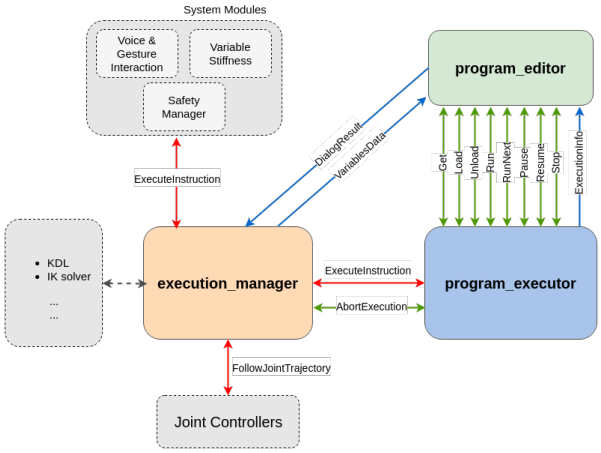

Fig. 1. FourByThree project architecture

II. Case Study

In the context of the FourByThree project in which the work presented here is inscribed, there are several industrial scenarios that include human-robot collaboration via natural communication. For an initial validation of the semantic multimodal interpreter, we have selected a scenario that involves two such collaborative tasks that are carried out via interaction between a person and a robot. One task involves the collaborative assembly/disassembly on the same dies, handling different parts of the dies and (un)screwing bolts as required. The other task involves a collaborative deburring operation of wax patterns that requires managing different parts adequately in order to build a mould.

In the case of the assembly task, the human and the robot work independently (un)screwing bolts on different parts of the die, and then they work together simultaneously (un)screwing different bolts on the same die cover. For the deburring activity, the human and the robot perform sequential tasks on the same workpiece in a sychronized manner, where the person glues and positions parts on the workbench while the robot deburrs them.

III. Related work

Over the last two decades, a considerable number of robotic systems have been developed showing Human-Robot Interaction (HRI) capabilities [3], [4]. Although recent robot platforms integrate advanced human-robot interfaces (incorporating body language, gestures, facial expressions and speech) [5], [6] their capabilities to understand human speech semantically remain quite limited. Endowing a robot with semantic understanding capabilities is a very challenging task. Previous experiences with tour-guide robots [7], [8] show the importance of improving human-robot interaction in order to ease the acceptance of robots by visitors. In Jinny's HRI system [8], voice input is converted to text strings, which are decomposed into several keyword patterns and a specialized algorithm finds the most probable response for that input. For example, two questions like 'Where is the toilet?' and 'Where can I find the toilet' are interpretedin the same way, since the keyword pattern of 'where' and 'toilet' are extracted from both cases.

Human-robot natural interactions have also been developed in industrial scenarios. For instance, Bannat et al. [2] introduced an interaction that consisted of different input channels such as gaze, soft-buttons and voice in an industrial scenario. Although voice constituted the main interaction channel in that use scenario, it was solved by command-word-based recognition.

SHRDLU is an early example of a system that was able to process instructions in natural language and perform manipulations in a virtual environment [9]. Researchers followed on that work towards extending SHRDLU's capabilities into real world environments. Those efforts branched out into tackling various sub-problems, including Natural Language Processing (NLP) and Robotics Systems. Notably, MAc Mahon et al. [10] and Kollar et al.[11] developed methods for following route instructions given through natural language. Tenorth et al. [12] developed robotic systems capable of inferring and acting upon implicit commands using knowledge databases. A similar knowledge representation was proposed by Wang and Chen [13] using semantic representation standards such as the W3C Web Ontology Language (OWL) for describing an indoor environment.

A generic and extensible architecture was described in [14]. The case study presented there included gesture and voice recognition, and the evaluation showed that interaction accuracy increased when combining both inputs (91%) instead of using them individually (56% in the case of gestures and 83% for voice). Furthermore, the average time for processing both channels was similar to the time needed for speech processing.

Our work is based on this extensible architecture, combining gesture and speech channels and adding semantic aspects to the processing.

IV. Multimodal Interaction Semantic Approach

The approach proposed in this work aims at creating a safe human-robot collaborative environment in which interactions between both actors happen in a natural way (understanding by 'natural' the communication based on voice and gestures). We propose a semantic multimodal interpreter prototype that is able to process voice and gesture-based natural requests from a person, and combine both inputs to generate an understandable and reliable command for industrial robots, enhancing safe collaboration. For such a semantic interpretation, we have developed four main modules, as shown in Fig. 2: a *Knowledge-Manager* module that describes and manages the environment and the actions that are feasible for robots in a given environment, using semantic representation technologies; a *Voice Interpreter* module that given a voice request, it extracts the key elements on the text and translates them into a robot-understandable representation, combining NLP and semantic technologies; a *Gesture Interpretation* module mainly for resolving pointing issues and some simple orders like stopping an activity; and a *Fusion Engine* for combining

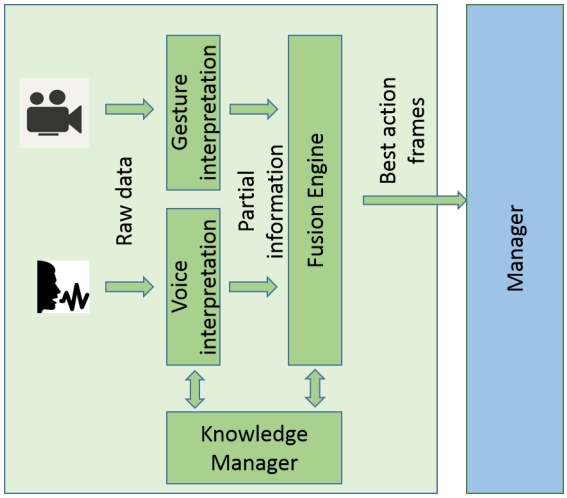

Fig. 2. Multimodal semantic approach architecture

the output of both text and gesture modules and construct a complete and reliable order for the robot.

These main modules are described in detail in the following subsections.

A. Knowledge Manager

The knowledge manager comprises ontologies that model environmental information of the robot itself, including its own capabilities. In addition, the knowledge manager allows modeling the relationships between the concepts. These relationships are implicit rules that can be exploited by reasoners in order to infer new information from the ontology. As a result, reasoners can work as rule engines in which human knowledge can be represented as rules or relations.

Ontologies have many practical benefits. They are very reusable and flexible at adapting to dynamic changes, thus avoiding to have to re-compile the application and its logic whenever a change is needed. Being in the cloud makes ontologies even more reusable, since different robots can exploit them, as was the case with e.g., RoboEarth [15].

Through ontologies, we model the industrial scenarios in which industrial robots collaborate with humans, in terms of robot behaviors, task/programs they can accomplish and the objects they can manipulate/handle from an interaction point of view. We distinguish two kinds of actions: actions that imply a status change on a robot operation, like *start* or *stop*, and actions related the robot capabilities such as *screw*, *carry*, *deburring* and so on.

Relations between all the concepts are also represented, which adds the ability for disambiguation during execution. This ability is very useful for text interpretation, since different actions can be asked from the robot using the same expression. For instance, people can use the expression *remove* to request the robot to *remove a burr*, but also to *remove a screw*, depending on whether the desired action is *deburring* or *unscrewing* respectively. If the relationships between the actions and the objects over which the actions are performed are known, the text interpretation will be more accurate, since it will be possible to discern in each case to which of both options the expression *remove* corresponds. Without this kind of knowledge representation, this disambiguation problem is far more difficult to solve.

For task/programs we make an automatic semantic extension exploiting WordNet [16] each time the robot is initialized. In this way, we obtain different candidate terms referring to a certain task, which is useful for text interpretation mainly, as it is described bellow.

For the current implementation, we have considered the two contexts described in the Case Study section. We have identified the possible tasks the robot can fulfill in both scenarios and we have created a knowledge base starting from the knowledge manager ontology. We have also included in the knowledge base the elements that take part in both processes, together with the relationships they have with respect to the tasks. This knowledge base is published in StarDog 4.0.5 Community version[17] and extended with WordNet as explained before.

B. Voice Interpreter

Given as input, a human request like *Remove the burrs from there* in which a person indicates (partially) the desired action via voice, the purpose of this module is to understand exactly what the person wants and if it is feasible to generate the necessary information for the robot. For instance, in the example just mentioned, the voice interpreter should deliver that the verb *remove* corresponds to the deburring action and check if it is a feasible action for the current collaborative robot. For such an interpretation, the module is divided into three main steps:

- The first step concerns to speech recognition.
- The second step is based on superficial information, in the sense that it does not take into account the meaning of words in the context. Its only purpose is to extract the key elements from the given order.
- The last step attempts to identify the action that is asked for, considering the key elements in the given context.

For speech recognition, we use Google Speech API[18]. Specifically, after recording the request of the operator, we send the audio file to Google Speech API obtaining the corresponding text.

Upon this text, in the second step, we apply natural language processing techniques using FreeLing, an open source suite of language analysis tools [19]. In particular, we apply a morphosyntactic analysis and dependency parsing to a set of request examples from different people. In this way, we obtain the morphosyntactic information of every element and about the request itself. We revise the complete information manually and identify the most frequent morphosyntactic patterns. From them, we extract elements denoting actions, objects/destinations (target onward) and explicit expressions denoting gestures, such as *there* and *that*. Following, we implement those patterns as rules, obtaining a set of rules that, given a FreeLing-tagged sentence, is able to extract the key elements on it.

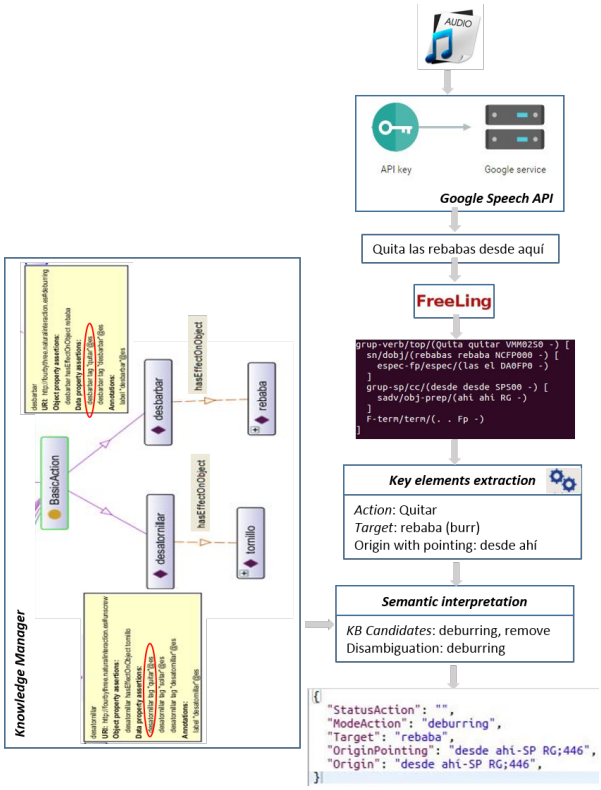

Fig. 3. Voice interpreter execution sequence

The aim of the second step is to identify which one of the tasks that the robot is able to perform suits the request best, considering the key elements in it. We undertake this step by making use of the knowledge base information described above. First, we verify if the identified actions are among the feasible tasks described in the knowledge base, and then we apply a disambiguation step using the target information, as explained before. This process results in the delivery of the best fits for the given input, from among the potential tasks obtained from the previous step.

The module output consists of frames, one for each potential task candidate, including information denoting gestures, if any exists.

Going back to the example about asking a robot to deburr a piece from a certain point on, the execution sequence of this module is detailed in Fig. 3 (*Remove the burrs from here*, example request in English).

Although it is not the case in the current example, since a single request can include different pointing gestures (e.g., *Take the piece from here to there*), the positions of the corresponding voice/text elements within the whole request are identified. These positions will be key in the fusion engine for aligning both voice and gesture inputs.

C. Gesture Interpretation

Two kinds of gestures are addressed within the *FourByThree* project: pointing gestures and gestures for simple commands such as stop/start. The case presented in this paper deals with pointing gestures that are recognized by means of point-cloud processing. In this context, the system must be able to not only recognize the pointing gesture, but also deliver within a certain period time how many different pointing gestures have occurred and which ones those are, in terms of *x,y,z* coordinates.

Fig. 4. Pointing gesture functional demonstrator

The initial setup consists of a collaborative robot and a sensor capable of providing dense point clouds, such as the ASUS Xtion sensor, the Microsoft Kinect sensor, or the industrial-grade Ensenso system by IDS. The sensor is placed above the human operator and orientated towards the working area of the robot, so that the point cloud obtained resembles what the human operator in perceiving in the working environment (see Fig. 4).

The point cloud is then initially divided into two regions of interest (ROI), the first one corresponding to the gesture detection area, and the second one defining the working area of the robot where the pointing gesture will be applied.

With this setup, two main problems need to be solved for the interaction between the person and the robot to succeed:

1) Robust estimation of the direction of the pointing gesture.
2) Intersection of the pointing gesture with the working area of the robot.

1) Robust estimation of the pointing gesture: The ROI for the pointing gesture detection is initially defined by specifying a cuboid in space with respect to the reference frame. In this case, the reference frame is the sensor frame, but it can also be defined using another working frame, provided a tf transformation exists between the frame used and the sensor frame. For robustness, the pointing gesture is defined using the forearm of the human operator. To identify the arm unequivocally, an euclidean cluster extraction is performed.

2) Intersection of the pointing gesture with the working area of the robot: The main objective of a pointing gesture is to determine the point on the working area that is being pointed at. To identify this point, the points in the cloud corresponding to the pointing line are selected, from the furthest one all the way to the origin of the line that corresponds to the pointing arm. For each one of the points, a small cuboid is defined, and the ROI of the working area of the robot is filtered with it. If more than N points of the working area are

present inside the small centered cuboid defined in the points of the projection line, an intersection has been found. The final intersection point that is published is the closest one to the origin of the projection line. As a threshold, a minimum euclidean distance value is defined in order to avoid detecting intersections corresponding to the proper point cloud of the arm that generates the pointing gesture.

When detecting gestures in a time frame, a spatial filtering approach has been implemented to distinguish among real stable pointing gestures and natural arm movements. The system is monitoring the intersection points obtained by the algorithm, and once a valid intersection point is obtained, the spatial filtering monitoring is launched. To detect a stable gesture, N consecutive intersection points must be contained in a defined cube whose centroid is the first intersection point obtained. The number of consecutive intersection points and the edge of the filtering cube are defined as parameters. A pointing gesture is considered stable and valid if it fulfills the previous explained condition. If not, the points of the last filtering operation are discarded. Valid points are queued during the time frame, and dispatched at the end of the acquisition time according to the format described below.

```
{"points": [
    {"x": "x1","y": "y1","z": "z1"},
    ... ,
    {"x": "xN","y": "yN", "z": "zN"}
]}
```

D. Fusion Engine

The fusion engine aims to merge both the text and the gesture outputs in order to deliver the most accurate request to send to the executive manager. The engine considers different situations regarding the complementary and/or contradictory levels of both sources.

As a first approach, we have decided the text interpreter output to prevail over the gesture information. In this way, when a contradictory situation occurs, the final request will be based on the text interpretation. When no contradiction exists between both sources, the gesture information is used either to confirm the text interpretation (redundant information), or to complete it (complementary information). For instance, using both voice and a gesture to stop a specific action provides redundant information through both channels. In contrast, using voice to determine an action and a gesture to indicate the location of the object that should suffer that action provides complementary information through both channels. In the second case, the knowledge base is used to check if the gesture information makes sense for a given task, discarding incoherent frame generation.

As a result, the fusion engine will send to the executive manager the potential, coherent and reliable requests that are understandable for the robot. The executive manager will then be in charge of task-planning issues considering those potential requests.

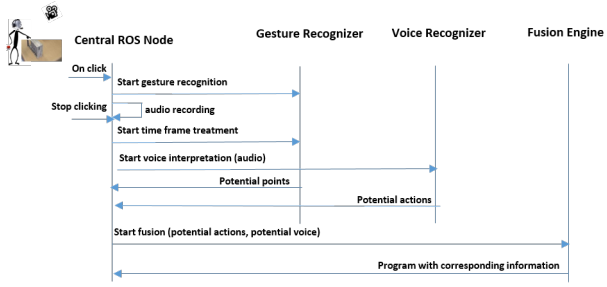

Fig. 5. Multimodal Interaction process sequence

In the current implementation, only the pointing gesture is included, and for that reason we have only tested the complementary functionality of the fusion engine. For the example we are affording, the output is the deburring program with the coordinates of the given point as parameter, that is sent to the execution manager who is in charge of managing when the request can be accomplished.

```
{"program": "deburring",
    "from": {"x": "x1","y": "y1","z": "z1"} ,
    "to":{}
}
```

V. Semantic Multimodal Interpreter in Action

This section reports on a set of tests of the Semantic Multimodal Interpreter with different types of requests comprising different voice and gestures inputs.

The initial setup consists of a button for triggering semantic interpreter, a microphone for voice input, a sensor capable of providing dense point clouds and a collaborative robot together with a human operator within a room in the context of the collaboration tasks introduced in the Case Study section.

Every time the operator wants to rise a requests for the robot, he should click the button and just keep it until the end of the request. When the system detects the click on the button, it starts recording the voice through the microphone and starts the time frame for the gesture module, who will be recognizing potential points until the operator stops clicking the button. Once it happens, voice interpreter is triggered with the recorded audio, and in parallel the gesture module starts the delivery of how many points have been pointed. When both modules returns the potential candidates in terms of actions and points, the fusion engine is triggered to estimate the full command to be sent to the manager with the corresponding information. The process sequence is shown in Fig. 5.

We have simulated the initialization process and the interpretation of some sample requests in order to validate the functionality. The initialization (mainly knowledge base creation) that happens together with the robot initialization takes around 15 seconds to complete.

Regarding the time required for gesture and voice interpretation, between 1 and 2 seconds are necessary for gesture recognition, while for voice interpretation times vary depending on

TABLE I
VOICE INTERPRETER DISAGGREGATED TIMES(SECONDS)

Voice Request	GSA	Freeling	Total
Quita ese tornillo (Remove that screw)	2	1	3.836
Quita esa rebaba (Remove that burr)	2	1	3.769
Empieza a atornillar la pieza (Start to screw the piece)	2	1	4.384
Comienza el mecanizado de la pieza redonda de allí (Begin with the machining of the round piece that is there)	3	1	5.749
Desatornilla de aquí a allí (Unscrew from here to there)	2	1	4.031
Detén el mecanizado (Stop the machining)	2	1	4.107

the complexity and length of the request (typically between 3.5 and 6 seconds). As it is shown in Table I, one of the most critical steps within the voice interpreter module is the Google Speech API (GSA in TableI) that takes around 2-3 seconds to process each petition. For FreeLing, the response times remain stable for all the examples, whilst the required time for the identification of the key elements varies depending on the amount of elements to manage and if the disambiguation step is required.

We are aware of the current high execution time required for interpreting a natural request when thinking in a human-robot collaboration environment. The main reason for such a high execution time is the sequential execution of different third-party tools such as Google Speech API and FreeLing. We plan to work on reducing it as much as possible once we conclude the evaluation we are currently carrying out in a laboratory environment.

VI. CONCLUSIONS AND FUTURE WORK

We have presented a semantic-driven multimodal interpreter for human-robot collaborative interaction focused on industrial environments. The interpreter relies on text and gesture recognition for request processing, dealing with the analysis of the complementary/contradictory aspects of both input channels, taking advantage of semantic technologies for a more accurate interpretation due to the reasoning capabilities it provides.

This approach is generic and it can be applied in different industrial scenarios. However, in order to evaluate this approach, we are working on a specific scenario that includes the human-robot collaborative activities of assembling and deburring. We intend to measure the whole system accuracy as well as the benefit of a multimodal system against a monomodal one in industrial environments. In addition, we will assess the usability and the benefits of such a system in industrial scenarios, as part of the advancement towards natural communication in human-robot collaborative work.

ACKNOWLEDGMENT

The FourByThree project has received funding from the European Union's Horizon 2020 research and innovation programme, under grant agreement No. 637095

REFERENCES

[1] http://fourbythree.eu/.
[2] A. Bannat, T. R. J. Gast, W. Rösel, G. Rigoll, and F. Wallhof, "A multimodal human-robot-interaction scenario: Working together with an industrial robot," pp. 303–311, 2009.
[3] T. Fong, R. Illah, and K. Dautenhahn, "A survey of socially interactive robots." pp. 143–166, 2003.
[4] M. Goodrich and A. Schultz, "Human-robot interaction: A survey." pp. 203–275, 2007.
[5] P. G. H. H. K. N. R. Stiefelhagen, C. Fugen and A. Waibel, "Natural human-robot interaction using speech, head pose and gestures," pp. 2422 – 2427 vol.3, 2004.
[6] B. Burger, I. Ferrane, and F. Lerasle, "Towards multimodal interface for interactive robots: Challenges and robotic systems description," 2010.
[7] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers., F. Dellaert, D. Fox, D. Hähnel, G. Lakemeyer, C. Rosenberg, N. Roy, J. Schulte, D. Schulz, and W. Steiner, "Experiences with two deployed interactive tour-guide robots," 1999.
[8] K. Gunhee, C. Woojin, K. Munsang, and L. Chongwon, "The autonomous tour-guide robot jinny," pp. 3450–3455, 2004.
[9] T. Winograd, "Procedures as a representation for data in a computer program for understanding natural language," 1971.
[10] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language, knowledge, and action in route instructions." 2006.
[11] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward understanding natural language directions." pp. 259–266, 2010.
[12] M. Tenorth, L. Kunze, D. Jain, and M. Beetz, "Knowrob-map - knowledge-linked semantic object maps," 2010.
[13] T. Wang and Q. Chen, "Object semantic map representation for indoor mobile robots," pp. 309–313, 2011.
[14] S. Rossi, E. Leone, M. Fiore, A. Finzi, and F. Cutugno, "An extensible architecture for robust miultmodal human-robot communication," pp. 2208–2213, 2013.
[15] D. Di Marco, M. Tenorth, K. Hussermann, O. Zweigle, and P. Levi, "Roboearth action recipe execution," pp. 117–126, 2013.
[16] A. Gonzalez-Agirre, E. Laparra, and G. Rigau, "Multilingual central repository version 3.0: upgrading a very large lexical knowledge base," 2012.
[17] http://stardog.com/.
[18] https://console.developers.google.com/apis/api/speech/.
[19] L. Padró and E. Stanilovsky, "Freeling 3.0: Towards wider multilinguality," 2012.