Agent-based Reconfigurable Natural Language Interface to Robots
Human-Agent Interaction using Task-specific Controlled Natural Languages

Tamás Mészáros and Tadeusz Dobrowiecki
Department of Measurement and Information System,
Budapest University of Technology and Economics, Budapest, Hungary

Keywords: Human-Machine Interaction (HMI), Human-Robot Interface (HRI), Human/Robot-Communication (HCL), Controlled Natural Language (CNL), Robot Operating System (ROS).

Abstract: We present the architecture of a flexible natural language based interface to robots involved in tasks in a mixed-initiative human-robot environment. The designed interface uses a bidirectional natural language communication pertinent to the user, the robot and the tasks at hand. Tasks are executed via agents that can communicate and engage in conversation with the user using task-specific controlled natural languages. The final interface language is dynamically composed and reconfigured at the interface level according to the tasks and skills of the robotic system. The multi-agent infrastructure is fused with the ROS robotic middleware to provide seamless communication to all system components at various level of abstraction.

1 INTRODUCTION

The evolution of robots from strict manufacturing to various service domains widened the spectrum of the potential human collaborators to humans not versed in the details of the robotic systems.

In such mixed robot-human teams (let it be an ailing elderly and the household robot, robotic museum guide and its visitors, shopping mall guide and customers, etc.), the principal communication modality is the natural human language (NL), getting in addition around the reading and typing by assuming speech modality (Burgard 1999, Rousseau 2013, Khayrallah 2015). Such level of interaction does not demand from the human agent any additional professional knowledge and skills beside the natural knowledge of the task and goals.

Using natural language (i.e. speech) alone was already considered and serious reviews are available (Bastianelli 2014, Schiffer 2012). The natural language interactions were also discussed in the context of service robotics, ranging from simple canned commands, via speech act protocols to the full natural language conversation without any constraints and limitations (Fong 2003).

Plenty of research has already been done in translating human NL demands into commands interpretable by the robot, even to the low level of the ROS commands (Lauria 2002, MacMahon 2006, Kemke 2007, Tellex 2011, Schiffer 2012, Howard 2014, Bastianelli 2014, Ferland 2014, Stenmark 2015). These approaches however addressed applications where the robot had a number of well-defined configuration and fixed set of tasks for which natural languages were designed, and only a few considered the option (and the potential) of the robot feed-back in natural language (Green 2009) or more complex dialogues with humans (Huber 2002).

A fully bi-directional and adaptive NL-based human-robot interface (HRI) requires an architecture that is more advanced. The grammar and the vocabulary of the human-to-robot NL interaction should dynamically adapt to the specifics of the robot design and the changing set of tasks and situations that the robot encounters. In order to achieve this our approach was to design a system that can dynamically change the natural language used on the HRI by seamlessly integrating task-specific sublanguages used by individual components (agents) in the robotic system.

It is almost impossible to tackle this task in a uniform, generic and formal way. Some components admitting an informal, ad hoc design must also be included in the HRI architecture and design. In the present paper we propose an architecture constraining the communication to the level of so called controlled natural language (Wyner 2009). Our aim here is to keep the format of the conversation well acceptable

MÃ©szÃ¡ros T. and Dobrowiecki T.
Agent-based Reconfigurable Natural Language Interface to Robots - Human-Agent Interaction using Task-specific Controlled Natural Languages.
DOI: 10.5220/0006205306320639
In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence (ICAART 2017)*, pages 632-639
ISBN: 978-989-758-220-2

to the human, yet formally treatable by the robotic system to be used also internally between the distributed system components.

The context of the development is an interesting mixed-initiative application domain (described in detail in Section 3), where the robot acts as a guide to the human, and where the human acts as the helper to the robot, essentially acting for its costly gripping and sensing. It makes the design of the robot simpler (and permits to move more intelligence and resources to other system components), and extends the spectrum of the manageable objects (as a gripper human arm and hand outperforms the robotic technology in the variety of objects it can handle with gentleness and precision, in size, weight, shape, texture, etc., grasping, stacking, turning, placing, ... all with the same "gripper").

Our contribution here aims at the principled development of a bidirectional and flexible conversation interface spanning the full robotic system, from the hand held devices to the ROS nodes in the robot middleware.

2 ROS-BASED ROBOTS

2.1 Robot Operating System

ROS based system implementation means a collection of loosely coupled, independent computing entities (nodes) exchanging information via asynchronous messages. The ROS Master is that part of ROS, which is responsible for providing topical connectivity ("yellow pages") among the nodes.

ROS system can be deployed across multiple machines, with one of them designated to host the master. A node is a running instance of a ROS program. (Well-designed) ROS nodes are loosely coupled, with (generally) no programmed-in knowledge about other nodes in the architecture.

ROS nodes communicate with each other using a publisher-subscriber model: they send messages to a declared topic (publishing) and receive on demanded topic (subscribing). A node that wants to share information will publish messages on the appropriate ROS topic and a node that wants to receive information will subscribe to that topic. The ROS Master ensures that nodes will find each other. The messages themselves are sent directly from publisher to subscriber.

2.2 Reconfigurable Robots

One of the essential so called core robotic system abilities (beside adaptability, interaction, motion, manipulation, perception, cognitive ability, dependability, and decisional autonomy) is the configurability. It means the ability of the robot to be configured to perform a task or reconfigured to perform different tasks (H2020). Configuration may range from a simple change of parameter values, through re-programming the system, up to being able to alter the physical structure of the system (especially its body of sensors and actuators).

The configuration ability can be the prerogative of the system designer, but can be moved into the hands of the user, or even the robot itself. Among many possible mechanisms of configuration (configuration files, skilled operator interaction, unskilled operator interaction, automatic arrangement, remote communication of configuration, see (H2020)) the unskilled operator case is for us of special importance, as such operator (plain user) required usually an unprofessional and natural interface to interact with the robotic system, where the speech modality offers itself at once.

In the next section we will see that the intelligent trolley (our prototype) is configured from the buying user to the buying user, being effectively re-programmed by the user's shopping list and in the real-time by the course of conversation with the customer. The developed natural language interface (configurable linguistically in itself) is thus a tool in the hands of the user to configure the general purpose intelligent robot to become his or her personalized service robot, at least for the timespan of buying.

3 THE PILOT SCENARIO

Our prototype robot called iTrolley works in a self-service warehouse environment where its main task is to help customers to collect (large) boxes of selected wares. A typical application is a furniture store where the user selects goods in the display area and picks them up in the self-service storage area.

The iTrolley provides intelligent, automated and easy-to-use transport support. It helps the user to traverse the storage area along an optimal path while collecting goods. It answers questions and queries about the shopping and the goods, resolves problems and warns about mistakes during the entire process. The robot may also provide additional services like payment preparation, customer support related to the wares, marketing, etc.

3.1 Roles in the Pilot Scenario

The human **Customer** selects the wares in the display area (creating the shopping list) then s/he picks up the selected goods boxed for delivery in the storage area. Finally, the Customer pays at the checkout area and puts the boxes into a car waiting in the shipping area.

The **Shopping App** runs on the Customer's mobile phone and helps browsing and selecting wares in the display area. It maintains the shopping list and helps the Customer to communicate with the robot.

The **iTrolley Robot** is a self-moving autonomous transport robot that helps the Customer to carry goods. It guides the Customer to the appropriate shelves in the storage area and gives instructions which boxes should be collected. It helps in detecting and resolving problems, provides payment information at checkout and brings the goods to the appropriate shipping location.

The **Warehouse Management System** provides inventory, payment, advertisement and other services and information to the Robot and the Customer.

3.2 The Shopping Scenario

The Customer enters the Warehouse, installs the Shopping App software on her/his phone or picks up a warehouse hand device with the application already installed, starts browsing the displayed goods (furniture, household appliances etc.) in the display area.

To buy an item s/he scans its code with the Shopping App (adding the item to the shopping list). After having finished the browsing s/he enters the Storage area and selects an available iTrolley Robot, and connects the Shopping App to the selected robot.

The iTrolley Robot talks with the Customer in natural language using speech recognition and synthesis. It plans the preliminary route along the Storage Area to pick the goods and executes it. It directs the Customer to the appropriate shelf where the item on the shopping list can be found, instructs the Customer to pick items (boxes) from the shelf and to put them on the robot’s transport platform, identifies and verifies the picked items (by weighting them or detecting barcodes present on the boxes) and warns the Customer about wrongly picked items. It also communicates with other iTrolley agents to resolve route conflicts, continuously modifies its route according to the global information about other iTrolley routes, traffic jams, etc. (cooperative planning).

When all goods are collected the iTrolley Robot provides pre-payment information to the cashiers at the checkout area, takes its load to an available cashier to proceed with the payment.

After the customer paid for the goods the cashier accepts the transaction and allows the Robot to proceed to the shipping area, collecting the warehouse's hand device from the Customer (if applicable). The iTrolley finally transports the goods to the shipping area.

3.3 Natural Language Communication

The iTrolley Robot communicates with the Customer in the spoken natural language. This makes it possible for the Customer to formulate queries and commands easily and to understand well the information provided by the robot. Situations presented in the shopping scenario may require many kinds of communication between the system and the Customer ranging from instructions given to or by the robot, through information sharing activities till a general chat. The reconfigurable nature of the robot, the various shopping list items and the varying needs of the Customer yield also many possible (and changing) conversation scenarios.

To take all these into account we designed a flexible and modular framework for natural language human-robot communication.

4 ARCHITECTURE

Our solution is designed as a ROS-based system meaning that the ROS platform provides the basic organization and communication capabilities throughout the entire system. It connects the human-machine interface to the robotic components as well as to other application specific modules.

4.1 Interface Components

The interface's internal architecture contains the following components (Figure 1).

The **Voice Agent** runs on a mobile device and contains a speech recognition module for detecting the user's input, a speech synthesis module for passing messages from the system to the user, a ROS communication module that connects the agent to the ROS-based system, and a dispatcher component that analyzes the user's input in order to determine the appropriate target agent that can process that input. In the pilot system this component is called the Shopping App, and it also contains a camera with barcode scanner as an input device.

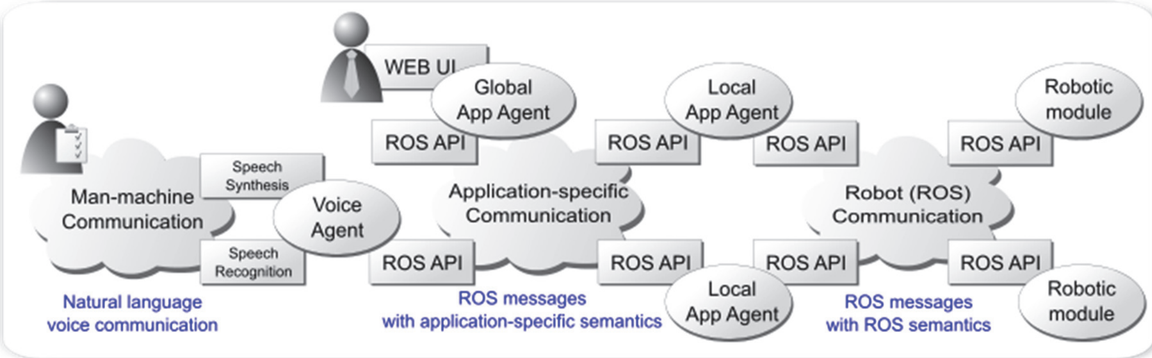

Figure 1: The high-level architecture of the system with various communication methods.

Robotic Modules run on the robot and provide services like physical movement and the navigation of the robot, handling sensor inputs and other hardware- and robot-specific modules. In the pilot scenario, these modules provide movement and navigation functions in the warehouse environment.

Local Application Agents run in close relation with the robotic modules. These agents provide higher level services like map discovery, route planning, error detection and recovery, etc. They may also possess natural language processing and understanding components, state management of human-robot conversations and application-specific functions. In the pilot scenario, these modules handle the shopping list, retrieve information (e.g. location) about the goods, plan the route in the storage area, coordinate item pick-up and perform task-specific natural language communication with the Customer.

Finally, **Global Application Agents** run independently from the robot and provide services like synchronizing the operation of multiple robots and resolving conflicts, logging and monitoring, etc. In our application, these agents provide warehouse inventory information and an administrator interface for monitoring the entire system.

4.2 Communication

All mentioned components communicate with each other using the ROS infrastructure, although they also encapsulate higher-level protocols in their messages. We differentiate the following three levels of communication.

At ROS level Robotic Modules communicate using standard ROS messages and semantics. These messages are delivered via well-defined topics and follow the standard semantics of the robotic components. For example, the navigation module communicates with the robot motor subsystem to reach the desired location in the storage area.

The Voice Agent, Local and Global Application Agents communicate among themselves using Agent Communication Languages (ACL) embedded in the ROS messages. This is the ROS-ACL level, which will be discussed in more detail in Section 5.3.

The Voice Agent communicates with the user using spoken natural language. In order to represent these messages internally the agent transforms them to ROS-ACL-NL messages. At this level natural language sentences are embedded in the ROS-ACL messages. These sentences are parsed and understood by Local and Global Application Agents that use task-specific controlled natural languages detailed later on in Section 6.

5 HOW DOES IT WORK

5.1 The Voice Agent

The main task of the Voice Agent is to communicate with the user using spoken natural language dialogues and transform them to ROS-ACL-NL messages. It provides no application-specific services or interface in general; it is merely the man-machine communication component of the system. In this respect it plays two roles: the Speaker and the Listener, for speaking and for speech recognizing. It can be placed on the robot itself or it can be deployed independently on a (mobile) device.

After starting up the Voice Agent implements two ROS topics (channels). The Speaker topic is used to receive requests for speech synthesis, while the Listener topic accepts subscription messages from other agents for notification about recognized input sentences from the Customer.

The Voice Agent's Speaker receives ROS-ACL-NL messages in which agents can request that natural language text present in the message should be spoken to the user. The Voice Agent queues the requests, and speaks them one-by-one using the output device. In our prototype implementation a simple FIFO scheduler is used for ordering the messages in the queue. If a given application would require it, other scheduling methods can also be used. It is even possible to use pre-emptive algorithms for urgent alert messages that notify the user of some imminent danger.

The Voice Agent working in Listener mode receives user input and routes it to the appropriate agent for processing. In order to do it, it accepts subscriptions for certain language patterns. These subscriptions are specified in ROS-ACL subscribe messages that include their ROS topic (to which the input should be delivered), the filter pattern for the input and a priority value for the given pattern.

Such patterns can be considered as language detection rules for task-specific controlled natural languages. An agent that wishes to communicate with the user provides these rules for the Voice Agent to select those input messages that it is interested in. Agents can specify or delete such rules dynamically as their operation (or life cycle) requires it. The Voice Agent always maintains a complete set of language rules according to the actual needs of other agents.

5.2 Task-specific Agents

Task-specific global and local application agents are responsible for services and functions required by the application. They communicate as ROS nodes in the system: they may create their own topics or subscribe to other topics as well. In addition to the standard ROS semantics they may also use ROS-ACL messages when communicating with other agents, and the ROS-ACL-NL protocol for natural language communication.

Application agents may also use the Voice Agent's previously mentioned Listener and Speaker topics to communicate with the user. In this case they send their own language detection rules to the Listener and subscribe for the incoming messages. They receive them from the Voice Agent in ROS-ACL-NL format and they also send such messages to the Voice Agent's Speaker to be spoken to the user. Examples for such agents can be found in Section 7.3.

5.3 ROS-ACL Messages

Agents communicate with each other using Agent Communication Language (ACL). The underlying communication framework is provided by the Robot Operating System, therefore ACL communication is embedded in ROS messages. This developed communication method is called ROS-ACL.

Any kind of ACL speech acts and semantics can be used as needed by the given application. For application-independent functions we have selected a limited set of ACL messages. Application agents may communicate with other agents using the ACL semantics: they *request* a service from another agent (e.g. text to speech messages can be sent as a request to the Voice Agent), *subscribe* to events (e.g. user input at the Voice Agent), *inform* other agents about events (e.g. recognized input text is sent from the Voice Agent to subscriber), and so on.

6 CONTROLLED NATURAL LANGUAGE COMMUNICATION

In a traditional software system user interface elements created and handled by various application components are assembled into a single graphical user interface by a compositing display manager.

The developed natural language interface implements a similar method. Every agent can communicate with the user using its own task-specific controlled natural language and a compositing interface (the Voice Agent) unifies these languages.

6.1 Natural Language Messages

When the user communicates with the system in spoken natural language the Voice Agent analyzes speech input and transforms it into ROS-ACL-NL messages where natural language sentences are embedded at the content level. These sentences are then received, parsed and understood by application agents.

Agents communicate with the user using task-specific controlled natural languages. These languages are created solely for the functions provided by the agents. E.g. in our pilot application we have created languages for shopping list assembly, item pick-up, etc. Since the system communicates with the user using a single voice input interface (the Voice Agent), it should be able to separate different languages, and it should determine a target agent for a given sentence from the human user. This is done by language detection rules provided by the application agents. The rules are used by the Voice Agent to determine the controlled natural language which is the best match for a given sentence and which agent is the target of that communication.

The rules have the following attributes: priority (importance), the ROS topic, and the language detection pattern.

Based on such rules the Voice Agent determines which patterns match a given input sentence, selects the topic with the highest priority, and assembles a ROS-ACL message to be sent to that topic.

In addition to the language detection, the Voice Agent is also able to maintain conversation between the user and the agents. Discourse detection is done using a similar rule-based scheme.

6.2 Conversation Management

To handle conversation between the user and the application agents we have applied finite state machines. These machines describe the possible states of the human-agent conversations and the possible state transitions defined by the application and the rules of the controlled languages.

The states of the finite state machine represent stages of the conversation with the user. A state can be changed by the user input (defined by the transitions for each state) or by the application logic of the agent (e.g. the agent can change its state when a message is received from another agent or ROS node). State transitions also specify the output to be sent to the user when the transition happens.

Every agent maintains its own state of conversation and provides information about allowed state transitions (possible user inputs specified via filter masks) to the Voice Agent. This interface agent collects all state transition filter masks and uses them in a similar way as language detection rules work.

There are also other mechanisms to ease the development of complex state machines (like “shortcut” transitions, initialization messages at states, etc.) that are not described in this paper.

7 PROTOTYPE TESTING

The demo application is implemented as a ROS-Java-Android hybrid system following the previously detailed system design and the pilot application scenario.

The ROS core of the pilot application runs in a simulated environment using Gazebo graphical robot simulator (gazebosim.org). It uses Jackal ROS package for navigation. We have developed several application-specific agents in Java to perform various tasks in pilot application. The Voice Agent is implemented in the Android ecosystem.

The software demonstration is based on open source tools and can be installed on Ubuntu using a Docker-based automated installation utility. The source code of our prototype application can be found at our project Web page (R5-COP 2016).

7.1 The ROS Simulator

We used the Gazebo simulator for testing and demonstration purposes.

It is a universal robotic simulation toolbox with advanced graphics and physics engine. Figure 2 shows the simulated robot and the storage area of the warehouse. For the sake of simplicity, Jackal robot model is used. The products (boxes) in the warehouse are located on standard shelves.

In addition to the visual interface the ROS simulator provides also several robotic components for sensor input, robot movement, route planning and navigation etc. These are used by application agents to perform various tasks in the pilot scenario.

Figure 2: The simulated storage area with the robot.

7.2 The Voice Agent Android App

We developed the Voice Agent as an Android application using the Google Speech Synthesis and Speech Recognition APIs to implement the necessary input and output function for the Listener and Speaker roles. The application has a simple configuration interface to specify the address of the ROS core.

After the configuration is finished the Voice Agent connects to the ROS core. When other agents in the system are connected to the Voice Agent, their detection rules are displayed for debugging purposes.

The application also contains a camera module to scan and recognize QR codes that are used to assemble the shopping list in the display area of the warehouse. We also used them to simulate the robot's camera that checks the collected wares during the pick-up phase in the storage area. In the simulation the robot asks the customer to scan the code of the selected box using the hand device, and it warns the user if a wrong box is picked.

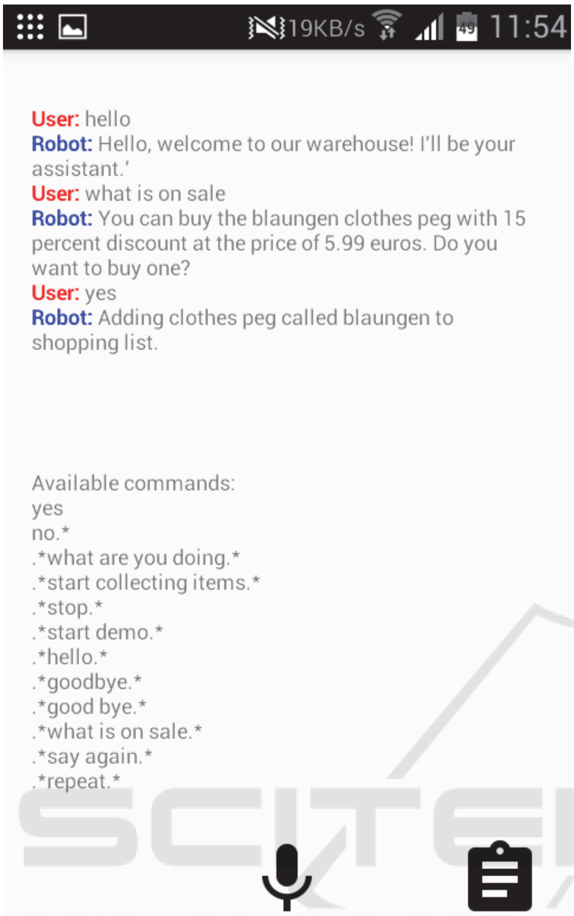

Figure 3: The Voice Agent as an Android application.

7.3 Application Agents

We have developed several application agents for the pilot scenario.

The simplest is HowAreYou Agent that performs general chat like greeting, goodbye and the like. It also has a wild-card rule with low priority that allows the system to respond to messages not recognized by any other agent.

The ShoppingList Agent is responsible for maintaining the state of the shopping list shown to the user on Android device. It also maintains the states of the wares, whether they are collected or not in the Storage Area, or if the robot is waiting directly in front of the shelf where the box can be found. It communicates with the ItemCollector Agent and with Pickup Agent.

The ItemCollector Agent is responsible for navigating the robot to items found on the shopping list, while the Pickup Agent instructs the user to pick up the box when the robot arrives to the appropriate place. This agent also validates the user's action by checking the QR code of the box.

There are other agents that communicate with the user in controlled natural language. For example, the Sales Agent advertises items on sale. All these agents have their simple controlled natural language rules, conversational states and state transitions.

In addition to these application agents the system also possesses a main agent, responsible for starting and stopping the execution of the demo application, the and the ROSDisplay Agent with a graphical user interface that provides monitoring of the system.

8 CONCLUSIONS

The developed system provides a natural language voice interface for a robotic system that provides a complex functionality to the human user. The internal complexity of the robotic system is tackled using several application-specific agents. These agents use task-specific controlled natural languages. The system uses a common natural language interface to communicate with the user using speech recognition and voice generation. This interface (the Voice Agent) uses language and discourse detection rules to identify the agent to which a given user message should be delivered.

The system is developed using the industrial standard ROS framework. Messages between agents and other ROS components are embedded into ROS-messages with different level of abstraction. The speech input and generation interfaces are implemented on the Android platform that provides readily available functionality for these purposes. The software running on the Android device is also connected to the ROS-based robotic system.

Although the system was developed for the iTrolley pilot scenario, it can be easily adapted to other applications as well. The communication framework and the Voice Agent are application independent thus need no customizations. This also implies that a developed system can also be reconfigured if the robotic hardware changes. In this case new ROS modules and modifications in Application Agents might be required depending on the nature of the change.

The developed solution not only accepts spoken commands from the users but it is also able to initiate and engage in conversations with the user in many different topics at the same time. This is an important feature for robots in application fields like health care, ambient assisted living. In these areas they not only perform, fulfil user commands (demands), but must direct, command, influence the users towards goals set by the staff controlling the user environment.

Furthermore, by replacing the ROS framework with other means for communication our approach may also be applied to other application areas where a multi-agent system should use a common human-machine interface to communicate with its users.

Other modalities could also be developed in addition to the speech interface. By extending the functionality of the Voice Agent other input methods like text input with autosuggest function, gesture recognition or simple menu-based interfaces could also be implemented. Such additional modalities can amplify the disambiguation capabilities and improve the efficiency of the communication (Green 2009, Breuer 2012, Fardana 2013).

ACKNOWLEDGEMENTS

This work was supported in part by ARTEMIS Joint Undertaking and in part by the Hungarian National Research, Development and Innovation Fund within the framework of the Reconfigurable ROS-based Resilient Reasoning Robotic Cooperating Systems (R5-COP) Project.

The prototype application was developed by a group of researchers including István Engedy, Péter Eredics and Péter Györke at our department.

REFERENCES

Bastianelli E., G. Castellucci, D. Croce, R. Basili, D. Nardi, 2014. Effective and Robust Natural Language Understanding for Human obot Interaction, *ECAI 2014, T. Schaub et al. (Eds.).*

Breuer, Thomas, et al. 2012. Johnny: An autonomous service robot for domestic environments. *Journal of intelligent & robotic systems* 66.1-2 (2012): 245-272.

Burgard W., A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, S. Thrun, 1999. Experiences with an interactive museum tour-guide robot, *Artificial Intelligence 114 (1999) 3–55.*

Fardana A.R., S. Jain, I. Jovancevic, Y. Suri, C. Morand and N.M. Robertson, 2013. Controlling a Mobile Robot with Natural Commands based on Voice and Gesture, *Workshop on Human Robot Interaction (HRI) for Assistance and Industrial Robots, VisionLab, Heriot-Watt University, 2013.*

Ferland F., R. Chauvin, D. Létourneau, F. Michaud, 2014. Hello Robot, Can You Come Here? Using ROS4iOS to Provide Remote Perceptual Capabilities for Visual Location, Speech and Speaker Recognition, *Proc. HRI '14 the 2014 ACM/IEEE Int. Conf. on Human-Robot Interaction, Pages 101-101.*

Fong T., I. Nourbakhsh, K. Dautenhahn, 2003. A survey of socially interactive robots, *Robotics and Autonomous Systems 42 (2003) 143–166.*

Green A., 2009. Design and Evaluating Human-Robot Communication, *PhD, KTH, 2009.*

H2020 – Robotics 2020 – Multi-Annual Roadmap For Robotics in Europe - http://www.eu-robotics.net/

Howard T.M., I. Chung, O. Propp, M.R. Walter, and N. Roy, 2014. Efficient Natural Language Interfaces for Assistive Robots, *IROS 2014 Workshop on Rehabilitation and Assistive Robotics, Sept 14-18, 2014, Chicago.*

Huber A., Bernd L. 2002. Users Talk to their Model Trains: Interaction with a Speech-based Multi-Agent System. *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2002, July 15-19, Bologna, Italy,*

Kemke C., 2007. “From Saying to Doing” – Natural Language Interaction with Artificial Agents and Robots, *Ch 9, Human-Robot Interaction, (ed.) N. Sarkar, Sept 2007, Itech Education and Publishing, Vienna, Austria.*

Khayrallah H., S. Trott, J. Feldman, 2015. Natural Language For Human Robot Interaction, *Proc. of the Workshop on Human-Robot Teaming at the 10th ACM/IEEE Int. Conf. on Human-Robot Interaction, Portland, Oregon.*

Lauria S., T. Kyriacou, G. Bugmann, J. Bos, E. Klein, 2002. Converting Natural Language Route Instructions into Robot Executable Procedures, *Proc. 11th IEEE Int. Work-shop on Robot and Human Interactive Communication, pp. 223-228.*

MacMahon M., B. Stankiewicz, B. Kuipers, 2006. Walk the Talk: Connecting Language, Knowledge, and Action in Route Instructions, *Proc. AAAI'06 the 21st Nat. Conf. on Artificial Intelli. - Vol 2, pp. 1475-1482.*

R5COP 2016, http://r5cop.mit.bme.hu/

Rousseau V., F. Ferland, D. Letourneau, F. Michaud, 2013. Sorry to Interrupt, But May I Have Your Attention? Preliminary Design and Evaluation of Autonomous Engagement in HRI, *J. of Human-Robot Interaction, Vol. 2, No. 3, 2013, pp. 41–61.*

Schiffer S., N. Hoppe, and G. Lakemeyer, 2012. Natural Language Interpretation for an Interactive Service Robot in Domestic Domains, *J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 39–53, 2012.*

Stenmark M., J. Malec, 2015. Connecting natural language to task demonstrations and low-level control of industrial robots, *Workshop on Multimodal Semantics for Robotic Systems (MuSRobS) IEEE/RSJ Int. Conf. on Intelligent Robots and Systems 2015.*

Tellex S., T. Kollar, S. Dickerson, M.R.Walter, A. Gopal Banerjee, S. Teller, N. Roy, 2011. Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation, *Proc. of the Nat. Conf. on Artificial Intelli. (AAAI 2011).*

Wyner, Adam, et al. 2009. On controlled natural languages: Properties and prospects. *International Workshop on Controlled Natural Language.* Springer Berlin Heidelberg, 2009.