Reasoning on Communication Between Agents in a Human-Robot Rescue Team

Asil Kaan Bozcuoğlu[1], Fereshta Yazdani[1], Daniel Beßler[1], Bogdan Togorean[2] and Michael Beetz[1]
{asil, yazdani, danielb, beetz}@cs.uni-bremen.de
bogdan.togorean@student.utcluj.ro

***Abstract*— Humans and robots start to team up in rescue missions to overcome together the challenges arising in kinematics and locomotions by humans. In order to enhance the success of these heterogeneous teams, human operators should know how their robotic partners will behave under different conditions.**

In this paper we integrate the high-level plans of robots in a rescue team into the OPENEASE web application in order to reason about actions and behaviors of agents at different timepoints and different locations. By using already existing Prolog queries or the new ones that they create, people can ask questions such as *why*, *how* and *when* a robot has done a certain behavior. This kind of queries will be useful for operators to diagnose and to understand the behaviors of their partners. We show two different exemplary use cases in a human-robot team: In the first one, the robotic agent misinterprets the command and goes somewhere else. In the second one, it interprets the command correctly and is able to successfully reach the region-of-interest. By reasoning on these two cases, one can conclude which kind of commands can be misinterpreted by the robot.

I. INTRODUCTION

Robots are taking part in helping their human partners in critical missons such as extinguishing fires in forests and rescuing humans from dangerous situations. Here, the interests are going towards real world scenarios in which robots have to perform complex tasks together in the team. An example of such a real world scenario is presented in the project *SHERPA* [1]. This project aims at the interaction of mixed human-robot rescue teams in a hostile terrain where the human team leader has to interact with her robotic team in order to find injured persons. In such cases, the communication between agents should be as clear as possible in order to avoid from fatal casualties.

One way to accomplish a smooth communication, is that humans have to be more *preemptive*. In other words, humans should anticipate how robots will react to their commands in different circumstances. In this sense, investigating and diagnozing how robots will behave under different conditions can help for such kind of anticipation.

On the other hand, robots are still not very easy to access and play with for many people due to the factors such as expensiveness and safety reasons. In addition, the robot simulations are hard to setup and use for them because of the lack of programming skills and their complexity to install and use.

With the emergence of cloud-robotics applications such as [2][3][4], the cutting-edge robot plan/knowledge frameworks can also be reached over the web without a necessity to the installation. In OPENEASE [2], people can make Prolog queries in order to reason about kitchen experiments that the robots have done previously. Even people without any Prolog knowledge can choose different experiments and reason by using predefined Prolog queries that are inserted by developers.

In this paper, we investigate the collaboration between a human team leader and a quadcopter and the achievement of tasks in the heterogeneous team. A key objective in this paper is to reconstruct and comprehend the task execution based on the behaviors of the different agents. In order to achieve this goal we propose an approach of reasoning about robot activity descriptions in a cloud-based knowledge service with a heterogeneous team in a rescue application. The contributions of this paper as follows:

- we introduce previously presented systems based on the interaction of human-robot teams and on knowledge processing and algorithms for machine learning;
- we introduce a simulation-based rescue mission with a human team leader and a quadcopter in which we show exemplarily the exchange and process of information between the teammates;
- we introduce different experiment types and their results which will be added with a new set of Prolog queries, into OPENEASE;
- finally we show how these Prolog queries can be used by the human rescue team members to reason about their robotic partners behaviors in the past operations and simulations even without any Prolog knowledge.

II. RELATED WORKS

In real world scenarios, the demand of Human-Robot Interaction (HRI) is extremely high-scheduled. Working together as partners, exchanging information and assisting one to another to achieve common goals are key issues that must be addressed. One of the challenges is to provide human and robots with models of each other [6]. In recent years, many work have been focused on developing robots that work and interact directly with humans, as assistants or

[1]A. Bozcuoğlu, F. Yazdani, D. Beßler and M. Beetz are with the Institute for Artificial Intelligence, Universität Bremen, 28359 Bremen, Germany
[2]B. Togorean is with the Faculty of Automation and Computer Science, Technical University of Cluj-Napoca, 400604, Cluj, Romania

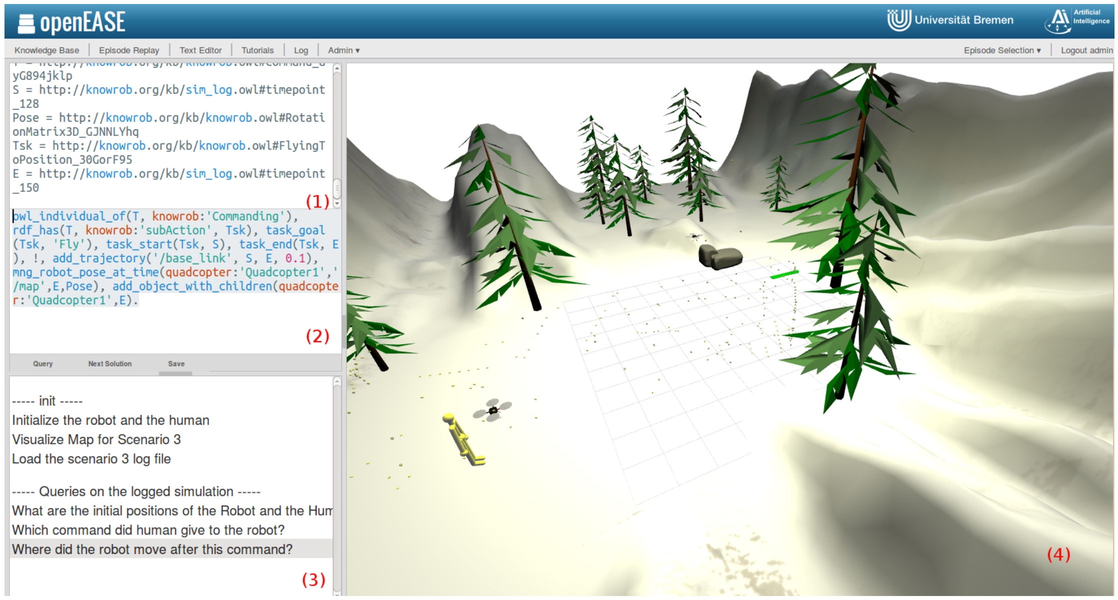

Fig. 2: The interface of OPENEASE. *The section (1)* is the Prolog console that one can see the previous queries and their results in the text form. *The section (2)* is the textbox that users can write new Prolog queries and execute them using *Query* button. In *the section (3)*, the predefined queries are listed. Users can query on these by clicking. In *the section (4)*, there is a 3D visual canvas. This canvas is updated when the users execute a Prolog query with a visual result.

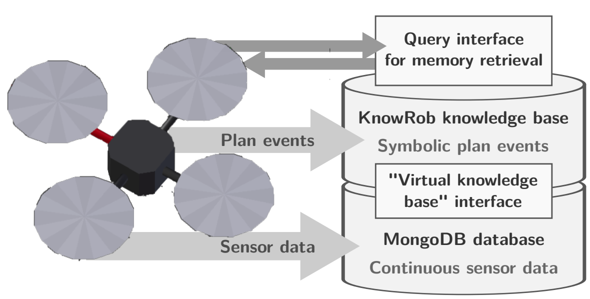

Fig. 1: The OPENEASE logging system. Image is based on [5].

teammates [7] [8] [9]. The field of Human-Robot Interaction research is addressed with communication, joint actions and human-aware execution that are challenging components and required by a smooth flow in human-agent activity [10]. In order to successfully accomplish common tasks robots require substantial knowledge about the environment they operate in and the objects they interact with. Such a knowledge system is offered in [11] that describes different kinds of knowledge and knowledge processing methods integrated in the system. A similar approach is done in [12] that works with the coordination of multiple robots in order to perform complex tasks assigned by people.

In the scope of robotics, there are some recent studies which put a special emphasis on knowledge processing. Saxena et al. [13], [14], introduce a learning methodology using natural languages in order to tell the robots how to accomplish a task. Moreover, there are some studies on using the world wide web as a deep knowledge source for robots for different goals such as concept learning [3] and task instructions acquisition [15].

Outside the scope of the robotics, Janowics et al. [16] propose a framework that combines machine learning algorithms with semantic web technologies. Wielemaker et al. [17] introduce a SWI Prolog-based web application similar to OPENEASE for the semantic web ontologies. They also introduce a SQL-like programming language, *SPARQL*, for researchers without Prolog knowledge.

In this paper we introduce a new approach, how robots' behavior can be better explored in order to enable a better communication in mixed human-robot teams. We investigate what challenges do arise when robots execute tasks in rescue missions and how these can be improved.

III. Proposed System

In the proposed system, the quadcopter uses the Cognitive Robot Abstract Machine (CRAM) [18] framework for planning that enables developers to define and to execute cognition-enabled plans on robots. Testing such a system in real life is difficult, because it is time consuming and associated with high costs. Therefore at the moment we are using the simulation as a development tool for simulating the team in a visual physical environment. We are using Gazebo [19] which is a multi-robot simulator for in- and outdoor environments. For creating the logs of experiments, we use the same logging mechanism described in [5]. In this mechanism, symbolic-level knowledge such as the tree of tasks inside the high-level plan, task parameters and failure and success states of each goal are recorded into the Web Ontology Language (OWL) format (Figure 1). This format is a knowledge representation language for ontologies that describes taxonomies and classification networks and is defining a whole structure of knowledge for various domains.

The low-level sensory data which includes the necessary links to high-level tasks are stored into files. These files use the data-interchange format JavaScript Object Notations (JSON) which is easy for humans to read and write and for machines to parse and to generate.

After the execution of tasks in Gazebo we start logging the experiments. These logs are directly integrated into OPENEASE whose Prolog engine KNOWROB [20] is fully-compatible with the used logging scheme. First, we put high-level logs onto the FTP server as experimental data which uses OPENEASE. Second, we import the JSON files into mongoDB instance of OPENEASE. Optionally, it is also possible to manually add some predefined queries into the query library.

In the end, by logging in OPENEASE web interface, users can select *Rescue Operations* experiment logs and query about details of them either using predefined queries or entering their own queries into the Prolog console. In Figure 2 is an illustration of the OPENEASE web interface indicated which visualizes the activities and the world state during the task execution at specific timepoints. An example of a query which is formulated in a high-level description in order to be comprehensible for humans can look like "where did the robot move after a specific command". The corresponding Prolog query would look like (Figure 2)

```
?- owl_individual_of(T, kr:'Commanding'),
   rdf_has(T, kr:'subAction', Tsk), !,
   task_goal(Tsk,'Fly'),
   task_start(Tsk, S),
   task_end(Tsk,E),!,
   add_trajectory('base_link',S,E,0.1),
   mng_robot_pose_at_time
   (quadcopter:'Quadcopter1','/map',E,Pose),
   add_object_with_children
   (quadcopter:'Quadcopter1', E).
```

An explicit description of the query is given in the next section. Additionally, it is also possible to add new predefined queries by external users if administrative rights are available.

IV. EXPERIMENTAL SETUP

In a typical SHERPA rescue scenario, there usually exists a rescue team consists of many agents and a human. In this paper we focus on a team consisting of a quadcopter and a human simulated in Gazebo in order to give a basic idea of the possibilities of this system. The communication between the team members is through commands given by the human operator. By commanding, the human operator points to an area that the quadcopter should inspect. Whenever the quadcopter finds an injured person in this area, it calls the human partner for help.

In order to show how useful these experiments can be used inside OPENEASE, we show two different human-robot communication scenarios in two different landscapes. In both scenarios, the human commands the robot to navigate to a particular area with different natural language instructions.

V. USE CASES

All of the queries explained in this section are also predefined in the corresponding experiment in OPENEASE with the natural language descriptions.

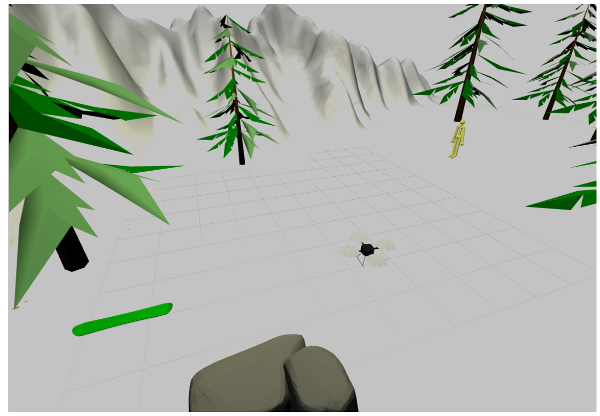

Fig. 3: The start positions of the agents in Figure V-A.

A. A Misunderstanding Case

In this case, a robot and a human are exploring a valley in the middle of mountains. In order to see the start position of these agents in order to be able to understand their behavior step by step, we can make a Prolog query as follows:

```
?- owl_individual_of(Exp, kr:'Experiment'),
   rdf_has(Exp, kr:'startTime', ST), !,
   add_stickman_visualization(xs:'Human1', ST),
   mng_robot_pose_at_time(q:'Q1','map',ST,P),
   add_object(q:'Q1',ST).
```

Afterwards, the positions of each agent is visualized and can be seen in the canvas (Figure 3).

To have a look into the command which the human gave, we execute the following query:

```
?- owl_individual_of(T, knowrob:'Commanding'),
   rdf_has(T, knowrob:'taskContext', Goal).
Goal = Navigate the area behind me
```

Finally, in order to see how the robot reacts and proceeds after getting the command "Navigate the area behind me", we will query the subtask of this corresponding *Commanding* task with the context *Fly*, then, we will look for the position of the robot at the end of this subtask:

```
?- owl_individual_of(T, knowrob:'Commanding'),
   subtask(T, Tsk), task_goal(Tsk, 'Fly'),
   task_start(Tsk, S), task_end(Tsk, E), !,
   add_trajectory('/base_link', S, E, 0.5),
   mng_robot_pose_at_time(q:'Q1','/map',E,P),
   add_object(q:'Q1',E).
```

As seen the final position of the robot in Figure 4, the robot has misinterpreted the command and gone in front of the human partner.

This example shows a misinterpretation and miscommunication between agents which can have fatal effects for the team and for the whole mission.

B. A Successful Communication Between Agents

In the second use case, again, a human team member and a robot are trying to find a victim in a valley passed by a river. When we make the same query for the agents' initial positions which we have also used in the first use case, we

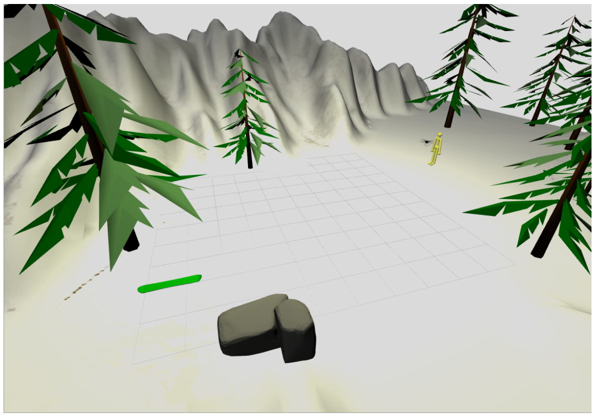

Fig. 4: The end positions of the agents in Figure V-A.

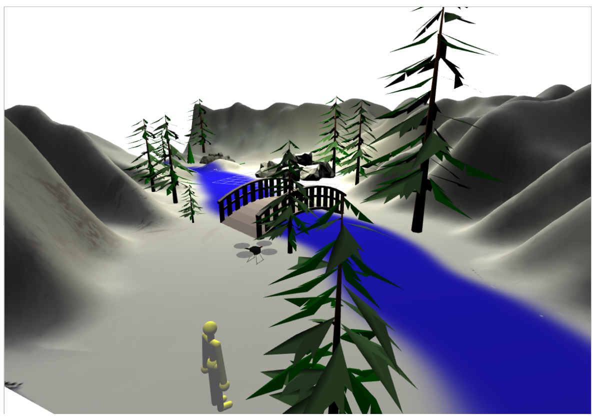

Fig. 5: The start positions of the agents in Figure V-B.

can see that the agents are standing close nearby the bridge (Figure 5).

One more time, for the command that the human gives we make a query with:

```
?- owl_individual_of(T, knowrob:'Commanding'),
   rdf_has(T, knowrob:'taskContext', Goal).
Goal = Explore 200 mt far away
```

If we look at the final position of the robot after using the command "Explore 200 mt far away", this time, we can see that the robot has successfully interpreted the command and gone to the region-of-interest (Figure 6).

In addition to the use cases, one essential result that a human team member can derive from these cases is, that a robot can successfully reach the region-of-interest when the given command includes an absolute position such as "Navigate the area that is 500 mt ahead". But if it includes some relative position definitions according to the team leader or to the robot itself, it is highly possible that the robot fails to accomplish the given command.

VI. Conclusion

In this paper, we have proposed a new experiment type to OPENEASE web application. The experiment was based on a collaboration with a human-robot team in a rescue scene in which the human member instructed the robot to

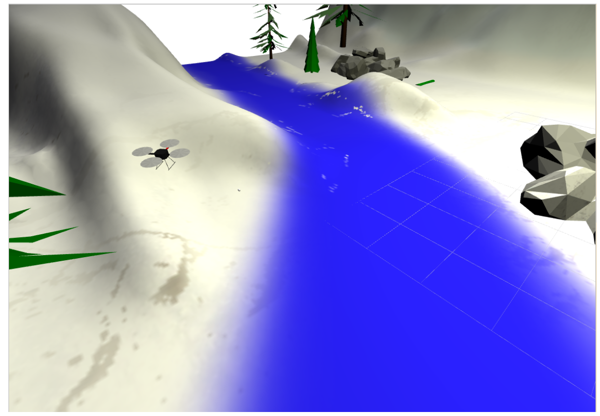

Fig. 6: The end position of the robot in Figure V-B.

look for injured persons in a scene. By using this kind of experiments, users, even without a technical background, can analyze, diagnose and debug behaviors of robots when they are commanded. In future, we are planning to extend the number of these rescue experiments with different scenarios so that humans can reason about the behaviors in an extended dataset in order to have a better anticipation and comprehension of the robot behaviors.

Acknowledgements

This work is supported by the European Commission's FP7 project, *SHERPA*, with grant number 600958.

References

[1] L. Marconi, C. Melchiorri, M. Beetz, D. Pangercic†, R. Siegwart, S. Leutenegger, R. Carloni, S. Stramigioli, H. Bruyninckx, P. Doherty, A. Kleiner, V. Lippiello, A. Finzi, B. Siciliano, A. Sala, and N. Tomatis, "The sherpa project: smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments," in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, College Station, Texas, USA, Nov. 5-8 2012.

[2] M. Beetz, M. Tenorth, and J. Winkler, "Open-EASE – a knowledge processing service for robots and robotics/ai researchers," in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, Washington, USA, 2015, finalist for the Best Conference Paper Award and Best Cognitive Robotics Paper Award.

[3] O. S. A. J. D. K. M. Ashutosh Saxena, Ashesh Jain and H. S. Koppula, "Robo brain: Large-scale knowledge engine for robots," arXiv, Tech. Rep., 2014, http://arxiv.org/abs/1412.0691.

[4] L. Riazuelo, M. Tenorth, D. D. Marco, M. Salas, L. Mösenlechner, L. Kunze, M. Beetz, J. D. Tardos, L. Montano, and J. M. M. Montiel, "Roboearth web-enabled and knowledge-based active perception," in *IROS 2013 Workshop on AI-based Robotics*, Tokyo, Japan, November 7th 2013.

[5] J. Winkler, M. Tenorth, A. K. Bozcuoglu, and M. Beetz, "CRAMm – memories for robots performing everyday manipulation activities," *Advances in Cognitive Systems*, vol. 3, pp. 47–66, 2014.

[6] T. B. Sheridan, "Eight ultimate challenges of human-robot communication," in *Robot and Human Communication, 1997. RO-MAN'97. Proceedings., 6th IEEE International Workshop on*. IEEE, 1997, pp. 9–14.

[7] N. Roy, G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Margaritis, M. Montemerlo, J. Pineau, J. Schulte, and S. Thrun, "Towards personal service robots for the elderly," in *Carnegie Mellon University*, 2000.

[8] T. Längle, T. Höniger, and L. Zhu, "Cooperation in human-robot-teams," 1997.

[9] I. R. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer, and A. Soto, “An affective mobile robot educator with a full-time job,” 1999.

[10] G. Klein, D. D. Woods, J. M. Bradshaw, R. R. Hoffman, and P. J. Feltovich, “Ten challenges for making automation a "team player" in joint human-agent activity,” *IEEE Intelligent Systems*, vol. 19, no. 6, pp. 91–95, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1109/MIS.2004.74

[11] M. Tenorth and M. Beetz, “KnowRob – Knowledge Processing for Autonomous Personal Robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 4261–4266.

[12] Y. Mori, Y. Ogawa, A. Hikawa, and T. Yamaguchi, “Multi-robot coordination based on ontologies and semantic web service,” in *Knowledge Management and Acquisition for Smart Systems and Services*, ser. Lecture Notes in Computer Science, Y. Kim, B. Kang, and D. Richards, Eds. Springer International Publishing, 2014, vol. 8863, pp. 150–164. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-13332-4_13

[13] Dipendra K Misra, Jaeyong Sung, Kevin Lee, Ashutosh Saxena, “Tell Me Dave: Context-Sensitive Grounding of Natural Language to Mobile Manipulation Instructions,” in *Robotics: Science and Systems (RSS)*. IEEE, 2014.

[14] Jaeyong Sung, Bart Selman, Ashutosh Saxena, “Synthesizing Manipulation Sequences for Under-Specified Tasks using Unrolled Markov Random Fields,” in *International Conference on Intelligent Robotics and Systems (IROS)*. IEEE, 2014.

[15] M. Tenorth, D. Nyga, and M. Beetz, “Understanding and Executing Instructions for Everyday Manipulation Tasks from the World Wide Web,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, USA, May 3–8 2010, pp. 1486–1491.

[16] K. Janowicz, F. van Harmelen, J. A. Hendler, and P. Hitzler, “Why the data train needs semantic rails,” *AI Magazine*, 2014.

[17] J. Wielemaker, W. Beek, M. Hildebrand, and J. van Ossenbruggen, “Cliopatria: A logical programming infrastructure for the semantic web,” *Semantic Web Journal*, 2015, under review.

[18] M. Beetz, L. Mösenlechner, M. Tenorth, and T. Rühr, “Cram – a cognitive robot abstract machine,” in *5th International Conference on Cognitive Systems (CogSys 2012)*, 2012.

[19] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, pp. 2149–2154 vol.3, 2004.

[20] M. Tenorth and M. Beetz, “KnowRob – A Knowledge Processing Infrastructure for Cognition-enabled Robots,” *International Journal of Robotics Research (IJRR)*, vol. 32, no. 5, pp. 566 – 590, April 2013.