Help Me! Sharing of Instructions Between Remote and Heterogeneous Robots

Jianmin Ji[1], Pooyan Fazli[2,3(✉)], Song Liu[1], Tiago Pereira[2], Dongcai Lu[1], Jiangchuan Liu[1], Manuela Veloso[2], and Xiaoping Chen[1]

[1] School of Computer Science and Technology, University of Science and Technology of China, Hefei, China
{jianmin,xpchen}@ustc.edu.cn
[2] Computer Science Department, Carnegie Mellon University, Pittsburgh, USA
mmv@cs.cmu.edu, tpereira@andrew.cmu.edu
[3] Electrical Engineering and Computer Science Department, Cleveland State University, Cleveland, USA
p.fazli@csuohio.edu

Abstract. Service robots frequently face similar tasks. However, they are still not able to share their knowledge efficiently on how to accomplish those tasks. We introduce a new framework, which allows remote and heterogeneous robots to share instructions on the tasks assigned to them. This framework is used to initiate tasks for the robots, to receive or provide instructions on how to accomplish the tasks, and to ground the instructions in the robots' capabilities. We demonstrate the feasibility of the framework with experiments between two geographically distributed robots and analyze the performance of the proposed framework quantitatively.

Keywords: Service robots · Sharing instructions · Collective robot learning · Robot-robot interaction framework

1 Introduction

Enabling remote and heterogeneous robots to share plans and instructions on various tasks assigned to them is an emerging field in artificial intelligence and robotics [7,20]. As current robots are far from being omniscient, the challenge is to develop robots that can recognize the missing knowledge autonomously and acquire it through alternative sources to accomplish tasks. This capability is especially crucial for service robots as they need to understand various user requests and provide services accordingly.

In this paper, we introduce a new framework, which allows remote and heterogeneous robots to share instructions on how to accomplish tasks. We demonstrate the feasibility of the framework with experiments between KeJia (Fig. 1(a)) and CoBot (Fig. 1(b)) and analyze the performance of the proposed framework quantitatively.

© Springer International Publishing AG 2016
A. Agah et al. (Eds.): ICSR 2016, LNAI 9979, pp. 786–795, 2016.
DOI: 10.1007/978-3-319-47437-3_77

(a) KeJia (b) CoBot

Fig. 1. KeJia and CoBot

KeJia [1], located in Hefei, China, is an intelligent domestic service robot, capable of interacting with humans in natural language and performing manipulation tasks in indoor environments autonomously. The robot was developed following a cognitive approach based on open knowledge available as semi-structured data.

CoBot [19], located in Pittsburgh, United States, is an autonomous service robot, capable of performing tasks and interacting with humans robustly in a multi-floor building, which has serviced and traversed more than 1000 km. The robot was developed following a novel symbiotic autonomy approach, in which the robot is aware of its perceptual, physical, and reasoning limitations and is able to ask for help from humans proactively.

2 Robot-Robot Interaction: Framework Overview

Figure 2 illustrates the structure of the proposed framework, which defines a protocol for remote robots to share instructions. An example of an interaction between two robots in this framework is as follows:

- Robot 1 receives a task from a human user in natural language.
- Robot 1 encodes the user task into a structured language and subsequently grounds it in its internal representation.
- Robot 1 fails to compute a plan for the assigned task.
- Robot 1 builds descriptions of the task and the environment it is located in and sends them to the interaction framework.
- The framework finds Robot 2, which is capable of providing instructions on the requested task for Robot 1.
- Robot 2, with or without using online resources, generates instructions for the task and transfers them to the interaction framework.

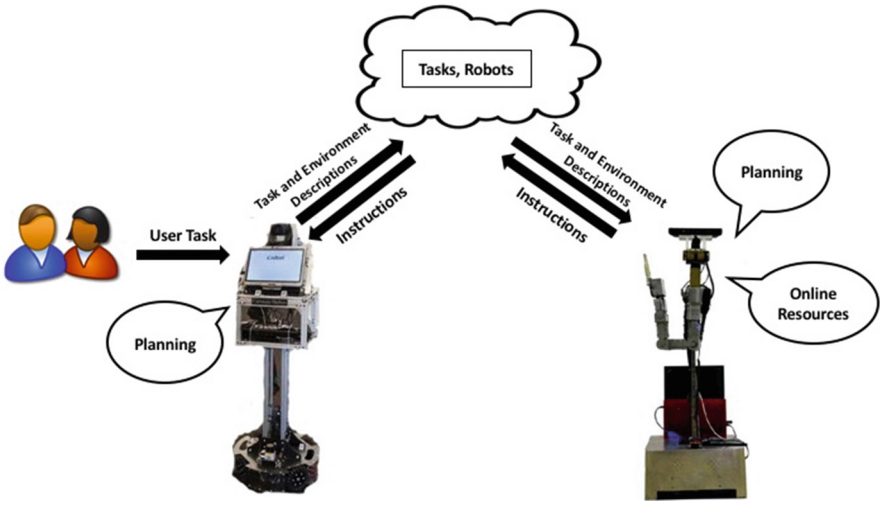


Fig. 2. The interaction framework between two robots.

– The framework saves the instructions for future queries about the task and transfers a copy to Robot 1.
– Robot 1 computes and executes a plan, using the received instructions to accomplish the task.

Section 4 discusses the details of the proposed robot-robot interaction framework, but before that, we introduce a common language to facilitate the cooperation between the robots.

3 Robot Communication Language

We introduce *Robot Communication Language (RCL)* to enable robots to share their knowledge and skills. RCL is a structured language to model tasks, environments, and instructions. The basic components of RCL include:

– *operation*: an operation that a robot should perform (e.g., *pick up, move, open*).
– *object*: an object, a human, or a location in the environment (e.g., *bottle, John, kitchen, living room*).
– *property*: a property of an object in the environment (e.g., *thirsty, closed, hot*).
– *relation*: a relation between two objects (e.g., *object A is on object B*).

The BNF definition of the basic components of RCL is as follows:

$$
\begin{aligned}
\langle \texttt{operation} \rangle &::= (\textit{operation}, \textit{object}) \mid (\textit{operation}, \textit{object}, \textit{object}) \\
\langle \texttt{property} \rangle &::= (\textit{property}, \textit{object}) \\
\langle \texttt{relation} \rangle &::= (\textit{relation}, \textit{object}, \textit{object}) \\
\langle \texttt{instruction} \rangle &::= \langle \texttt{operation} \rangle \mid \langle \texttt{sequence} \rangle \mid \langle \texttt{selection} \rangle \mid \langle \texttt{repetition} \rangle \\
\langle \texttt{sequence} \rangle &::= (\text{:seq } \langle \texttt{operation} \rangle^{+}) \\
\langle \texttt{selection} \rangle &::= (\text{:if } \langle \texttt{property} \rangle \text{ :then } \langle \texttt{instruction} \rangle) \mid \\
&\quad\ (\text{:if } \langle \texttt{relation} \rangle \text{ :then } \langle \texttt{instruction} \rangle) \\
\langle \texttt{repetition} \rangle &::= (\text{:while } \langle \texttt{property} \rangle \text{ :do } \langle \texttt{instruction} \rangle) \mid \\
&\quad\ (\text{:while } \langle \texttt{relation} \rangle \text{ :do } \langle \texttt{instruction} \rangle)
\end{aligned}
$$

Subsequently, we can define *tasks*, *environments*, and *instructions* in RCL as follows:

- (:task ⟨`operation`⟩): a task to be completed by a robot.
- (:env ⟨`property`⟩* ⟨`relation`⟩*): a model of the environment.
- (:inst ⟨`instruction`⟩): an instruction set.

4 Sharing Instructions Between Robots

4.1 Grounding the User Task

When a robot receives a task from a human user in natural language, it needs to map it to RCL. To this end, the robot uses the Stanford Parser [8] to generate the syntax tree and identify the grammatical structure of the input user request. Based on the parsing results, the robot fills corresponding phrases in ⟨`operation`⟩ (i.e., *operation* and *object*) for the assigned task. The next step is to ground *operation* and *object* into the corresponding symbols in the robot's internal representation. Synonyms in WordNet[1] are also used to facilitate the mapping process. A partial list of symbols used in the robots is shown in Table 1.

4.2 Asking for Help from a Remote Robot

If the robot fails to compute a plan for the grounded task, it needs to ask for help from a remote robot. To this end, the robot builds descriptions of the task and the environment in RCL and transfers them to the interaction framework. The framework matches the requested task with a robot capable of providing instructions for it.

Having received the descriptions of the task and the environment, the remote robot either computes a plan and builds an instruction set out of it, or instructions are extracted directly from online resources.

[1] http://wordnet.princeton.edu/wordnet/.

Table 1. A partial list of symbols used in KeJia and CoBot

Operation	**Description**
bring(O_1, O_2)	Bring object O_1 to location O_2
give(O_1, O_2)	Give object O_1 to human O_2
Object	**Description**
kitchen	Kitchen
fridge	Fridge
Property	**Description**
portable(O)	Object O is portable by the robot
holding(O)	Robot is holding the object O
Relation	**Description**
at(O_1, O_2)	Object O_1 is at location O_2
on(O_1, O_2)	Object O_1 is on object O_2

Generating Instructions from a Plan. The remote robot needs to compute a plan (i.e., a sequence of primitive actions) for the queried task. The environment is specified by the notion *state*, which is a set of predicates that are true in the environment. We define a *trajectory* as a sequence $\langle s_0, a_0, s_1, \ldots, a_{n-1}, s_n \rangle$, where s_i $(0 \leq i \leq n)$ is a state, a_j $(0 \leq j < n)$ is a primitive action, and s_{j+1} is the successor state of the state s_j after executing the action a_j. We can obtain the plan $\langle a_0, \ldots, a_{n-1} \rangle$ from such a trajectory.

To solve the planning problem, it is encoded into a logic programming language called *Answer Set Programming (ASP)* [11]. Each model of the ASP program corresponds to a possible plan for the task. The planning problem consists of an initial state, a goal state, and an action model. Action model is the set of primitive actions that are executable by the robot. Formally, a primitive action a is defined as a pair $\langle pre(a), eff(a) \rangle$, where $pre(a)$ and $eff(a)$ are preconditions and effects of a respectively.

The goal is to compute a sequence of primitive actions to get from the initial state to the goal state. Possible plans are computed by an ASP solver, iclingo [4]. A detailed description of this approach can be found in [6]. Having computed the plan, the robot encodes it into a set of ⟨`instruction`⟩s in RCL.

Generating Instructions from Online Resources. CoBot is capable of querying the web to learn the most probable location of an object in an indoor environment [13]. On the other hand, in KeJia, *Open Mind Indoor Common Sense (OMICS)* [9] is considered as a source of instructions. OMICS is an extensive database of common sense knowledge about home and office environments collected from non-experts over the web to enhance the capabilities of indoor autonomous robots. OMICS provides 11885 tables, including 28337 lines of instructions for 819 different user tasks. For some tasks, more than one

Table 2. An example table in OMICS

Task	Step number	Instructions
Fetch an object	0	Locate the object
	1	Go to the object
	2	Take the object
	3	Go back to where you were

instruction set has been defined in OMICS. Table 2 shows an example of the task "fetch an object" and the list of instructions to accomplish the task in OMICS.

Given a task, KeJia can query the OMICS database to find the closest match among the OMICS tables. Then, it extracts the instructions in the table for the queried task and encode them into ⟨`instruction`⟩s in RCL. Most instructions in OMICS are sequential. When the keywords "if", "then", "until", or "while" appear in the instructions, the ⟨`instruction`⟩ with the corresponding control structures is built in RCL.

4.3 Receiving Instructions from the Remote Robot

The robot grounds the instructions received in RCL into the corresponding symbols in its internal representation. It then computes a plan according to the instructions. The ASP solver, iclingo, is used to compute a trajectory with the least number of primitive actions. The planning procedure, considering the action model of the robot, computes a satisfying trajectory when the instructions miss some indispensable steps. For example, in OMICS, there are instructions on how to "get food from the fridge". The last two steps are "pick up food" and "close door". However, KeJia has only one arm, therefore it has to place the food somewhere before closing the fridge door.

5 Related Work

The RoboEarth project [20] offers a cloud robotics [7] infrastructure, which allows robots to share information and learn new skills and knowledge from each other. The RoboBrain project [14] aims at constructing a massive large-scale knowledge base for robots that learns from internet resources.

Understanding natural language commands has been regarded as a basic ability of a service robot [10,15,17]. Previous work have exploited instructions described in natural language to generate a plan for a user task [3,12,16]. There are also approaches for generating plans for household robots, using natural language instructions extracted from the web [18]. Also, important to our work is the capability to ground natural language phrases in the objects, actions, and relations in the target area [5].

(a) User asks KeJia to bring him a bottle of water.

(b) CoBot computes the most probable location to find a bottle of water.

(c) KeJia executes a plan based on the instructions received from CoBot.

(d) KeJia completes the task.

Fig. 3. Scenario 1: CoBot is helping KeJia.

6 Experiments and Evaluations

We propose an interaction framework, which allows robots to share instructions on how to accomplish tasks. We demonstrate the feasibility of the framework with experiments between two geographically distributed robots and evaluate the expressive power of Robot Communication Language (RCL) quantitatively.

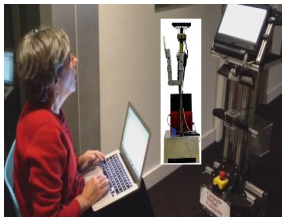
6.1 Experiments Between KeJia and CoBot

We use the rosbridge package [2] to connect KeJia and CoBot. The experiment consists of two scenarios.

In the first scenario (Fig. 3), the user asks KeJia: "Bring me a bottle of water". KeJia parses the request and maps it to RCL: (:task (bring, bottle of water)). KeJia then tries to compute a plan for the given task, but fails, as it does not know where to find a bottle of water. Hence, KeJia sends the failed subtask of finding a bottle of water to CoBot: (:task (find, bottle of water)).

CoBot grounds the task in its own symbols and querying the web, it computes the most probable location to find a bottle of water (i.e., dinner table) and subsequently generates the following instruction: (:inst (find, bottle of water, dinner table)). CoBot sends the instruction to the interaction framework and KeJia. KeJia grounds the received instruction in its own symbols and computes a plan to bring the bottle of water to the user.

(a) User tells CoBot that she is thirsty.



(b) KeJia suggests that CoBot give the user a bottle of water.

(c) CoBot executes a plan based on the instructions received from KeJia.

(d) CoBot completes the task.

Fig. 4. Scenario 2: KeJia is helping CoBot.

In the second scenario (Fig. 4), the user tells CoBot: "I am thirsty". CoBot does not know how to interpret the user request and since there is no task defined in the request, CoBot just builds a description of the environment, (:env (thirsty, human)), and sends it to the interaction framework and KeJia. KeJia uses OMICS to extract an instruction, that is, (:inst (give, bottle of water, human)), to satisfy the user desire and sends it to the interaction framework and CoBot. CoBot computes a plan based on the instruction to complete the task.

The framework enables both robots to assist each other in completing tasks that were unable to handle previously.

6.2 RCL Evaluation

OMICS recorded a great amount of instructions for everyday tasks written by internet users. Hence, we use it as a benchmark to evaluate the expressive power of RCL.

Table 3 shows the percentage of tasks, instructions, and tables (i.e., tasks and the instruction sets) in OMICS that were parsed and encoded into RCL successfully. As mentioned earlier, in OMICS, there are 11885 tables, including 28337 lines of instructions for 819 different user tasks. In summary, ∼58 % of the tables were parsed, and ∼28 % of the tables were converted to RCL successfully.

Table 3. Summary of the evaluations with OMICS

Test set	Tasks (819)	Instructions (28337)	Tables (11885)
Parsed	81.81 %	88.87 %	57.99 %
Converted to RCL	73.99 %	63.04 %	27.98 %

7 Conclusion and Future Work

We introduced a new framework, which allows remote and heterogeneous robots to share instructions on how to accomplish tasks that were unable to handle previously. We demonstrated the feasibility of the framework through experiments between KeJia, located in Hefei, China and CoBot, located in Pittsburgh, United States. We also introduced Robot Communication Language (RCL) to facilitate the collaboration process between the robots and evaluated its expressive power quantitatively. For future work, we intend to extend the framework to support multiple remote and heterogeneous robots. Various robots may generate different instructions for a user task and a robot can choose the most proper set of instructions among them.

References

1. Chen, X., Ji, J., Jiang, J., Jin, G., Wang, F., Xie, J.: Developing high-level cognitive functions for service robots. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), pp. 989–996 (2010)
2. Crick, C., Jay, G., Osentoski, S., Jenkins, O.C.: ROS and rosbridge: roboticists out of the loop. In: Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI 2012), pp. 493–494 (2012)
3. Dzifcak, J., Scheutz, M., Baral, C., Schermerhorn, P.: What to do and how to do it: translating natural language directives into temporal and dynamic logic representation for goal management and action execution. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2009), pp. 4163–4168 (2009)
4. Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Thiele, S.: Engineering an incremental ASP solver. In: Garcia de la Banda, M., Pontelli, E. (eds.) ICLP 2008. LNCS, vol. 5366, pp. 190–205. Springer, Heidelberg (2008)
5. Guadarrama, S., Riano, L., Golland, D., Gouhring, D., Jia, Y., Klein, D., Abbeel, P., Darrell, T.: Grounding spatial relations for human-robot interaction. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013), pp. 1640–1647 (2013)
6. Ji, J., Chen, X.: From structured task instructions to robot task plans. In: Proceedings of the 5th International Conference on Knowledge Engineering and Ontology Development (KEOD 2013), pp. 237–244 (2013)
7. Kehoe, B., Patil, S., Abbeel, P., Goldberg, K.: A survey of research on cloud robotics and automation. IEEE Trans. Autom. Sci. Eng. **12**(2), 398–409 (2015)
8. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003), pp. 423–430 (2003)

9. Kochenderfer, M.J., Gupta, R.: Common sense data acquisition for indoor mobile robots. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2004), pp. 605–610 (2003)
10. Kollar, T., Tellex, S., Roy, D., Roy, N.: Toward understanding natural language directions. In: Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI 2010), pp. 259–266. IEEE (2010)
11. Lifschitz, V.: Answer set planning. In: Gelfond, M., Leone, N., Pfeifer, G. (eds.) LPNMR 1999. LNCS (LNAI), vol. 1730, pp. 373–374. Springer, Heidelberg (1999)
12. Rybski, P.E., Yoon, K., Stolarz, J., Veloso, M.M.: Interactive robot task training through dialog and demonstration. In: Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI 2007), pp. 49–56 (2007)
13. Samadi, M., Kollar, T., Veloso, M.M.: Using the web to interactively learn to find objects. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI 2012), pp. 2074–2080 (2012)
14. Saxena, A., Jain, A., Sener, O., Jami, A., Misra, D.K., Koppula, H.S.: Robobrain: large-scale knowledge engine for robots. arXiv:1412.0691 (2014)
15. Shimizu, N., Haas, A.R.: Learning to follow navigational route instructions. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009), pp. 1488–1493 (2009)
16. Skubic, M., Perzanowski, D., Blisard, S., Schultz, A., Adams, W., Bugajska, M., Brock, D.: Spatial language for human-robot dialogs. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **34**(2), 154–167 (2004)
17. Tellex, S., Kollar, T., Dickerson, S., Walter, M.R., Banerjee, A.G., Teller, S.J., Roy, N.: Understanding natural language commands for robotic navigation and mobile manipulation. In: Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI 2011), pp. 1507–1514 (2011)
18. Tenorth, M., Nyga, D., Beetz, M.: Understanding and executing instructions for everyday manipulation tasks from the world wide web. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2010), pp. 1486–1491 (2010)
19. Veloso, M., Biswas, J., Coltin, B., Rosenthal, S., Kollar, T., Mericli, C., Samadi, M., Brandao, S., Ventura, R.: Cobots: collaborative robots servicing multi-floor buildings. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012), pp. 5446–5447 (2012)
20. Waibel, M., Beetz, M., D'Andrea, R., Janssen, R., Tenorth, M., Civera, J., Elfring, J., Gálvez-López, D., Häussermann, K., Montiel, J., Perzylo, A., Schießle, B., Zweigle, O., van de Molengraft, R.: RoboEarth - a world wide web for robots. Robot. Autom. Magaz. **18**(2), 69–82 (2011)