KeJia: The Integrated Intelligent Robot for RoboCup@Home 2013

Xiaoping Chen, Feng Wang, Hao Sun, Jiongkun Xie, Min Cheng, and Kai Chen

Multi-Agent Systems Lab., Department of Computer Science and Technology,
University of Science and Technology of China, HeFei, 230027, China
xpchen@ustc.edu.cn, {fenggew, hhsun, devilxjk, ustccm,
chk0105}@mail.ustc.edu.cn
http://wrighteagle.org/en/robocup/atHome

Abstract. This paper reports some recent progress on the project KeJia, whose long-term goal is aiming at the service robots with integrated intelligence. It focuses on the content ranging from the low-level hardware design to the high-level cognitive functions. These techniques and the integrated system have been tested in RoboCup@Home standard tests and other case studies.

1 Introduction

Recently more and more researchers from AI, Robotics and related areas, are showing their interest in intelligent indoor robots [1, 5–7, 15, 19]. There are three requirements are challenging them. Firstly, an intelligent indoor robot should be able to communicate with humans naturally. Secondly, it ought to possess some degree of autonomy, particularly, autonomously planning for tasks. Finally, it needs the capability of learning from its experience and humans and thus reach a higher performance; specifically, we hope the robot can acquire general knowledge through the human robot dialogue and other sources such as open knowlege bases.

The motivation of the project KeJia is attempting to develop intelligent indoor service robots that meet these three requirements. Several general-purpose approaches to meeting the requirements have been implemented in our robotic system KeJia for natural language processing [8], for hierarchical task planning [9], and for knowledge acquisition [7, 10]. We have tested these techniques and the whole system in RoboCup@Home league competitions from the year 2009 as well as other case-studies. In this paper, which serves as the team description paper of WrightEagle@Home for RoboCup@Home 2013, we concern ourselves with our latest research progress.

Section 2 gives an overview of our robotic system. The low-level functions of our Robot KeJia are described in Section 3. Section 4 specifies the details about the human-robot dialogue management and speech understanding. Section 5 elaborates a hierarchical approach to task planning. Finally we conclude in Section 6.

2 Architecture of KeJia

The hardware architecture of our robot KeJia was designed in 2012 and has shown its outstanding performance on the RoboCup@Home 2012 competition. Our robot is based on a two-wheels driving chassis. It is equipped with a lifting system that could adjust the height of its upper body quickly. A five degrees-of-freedom (DOF) arm makes our robot agile to fulfill the manipulating tasks in the indoor enviroments. It has a reach of over 83 centimeters and is able to hold a payload of up to 500 grams while fully extended. Our robot is about 1.6 meters height and weights about 40 kilograms. For supporting the real-time environmental perception, our robot is equipped with a Kinect camera, a high-resolution RGB camera, two laser range finders and a microphone. The 20AH battery makes a guarantee that our robot could be reliably running in continuous applications. The computational capability of KeJia is powered by the laptop setting on the back of robot. The image of our robot KeJia is shown in Fig. 1.

Fig. 1. The robot KeJia

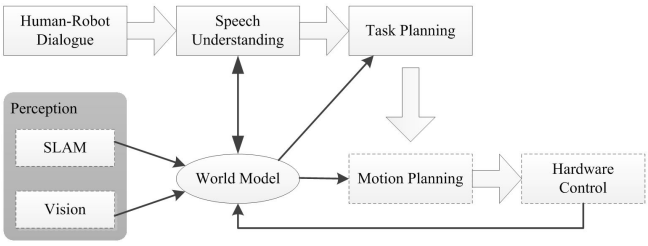

Fig. 2. Software architecture of KeJia

The software architecture of KeJia is shown in Fig. 2. Robot Operating System (ROS)[1] have been employed as the infrastructure supporting the commu-

[1] http://www.ros.org/wiki/

nication between modules in our robotic system. Our robot is driven by input from Human-Robot Dialogue module. Through the Speech Understanding module, the utterances from users are translated into the internal representations of the robot. These representations are in the form of Answer Set Programming (ASP) language [11] which is a Prolog-like logical language. An ASP solver is employed in the Task Planning module to automatically make decisions given the translated results. The Task Planning module then generates the high-level plans for users' tasks. The generated course of actions is fed into the Motion Planning module. Each action is designed as a primitive for KeJia's Task Planning module and could be carried out by the Motion Planning module and then autonomously executed by the Hardware Control module.

The changes of the external environments and the internal state of KeJia itself are perceived by the low-level modules (i.e., SLAM, Vision, and Hardware Control module) and used to update the World Model. The Motion Planning module deals with a repertoire of (low-level) routines and predefined parameters. For each low-level function of the robot, such as object recognition and manipulation, there is a routine, which involve uncertainties that could be best modeled with quantitative mathematical methods.

The integrated system of KeJia has been tested in RoboCup@Home league competitions in the past four years. We won the 2nd place in the RoboCup@Home 2011 and the 4th place in last year. The high-level cognitive functions have also been examined in a series of case studies[2]. At this point, KeJia have shown its competence in offering the general purpose service with incomplete or erroneous information, learning operations on a microwave oven through reading a manual, and acquiring open knowledge from spoken dialogue and from knowledge base.

3 Low-level Functions

3.1 Self-Localization and Navigation

For self-localization and navigation, a 2D occupancy grid map is generated first from the raw data collected by laser scanners through a round travel within the rooms aforehand [12]. Then the map is manually annotated with the approximate location and/or area of rooms, doors, furniture and other interested objects. Finally, a topological map is automatically generated, which will be used by the global path planner and imported as a part of prior world model. With such map, scanning match and probabilistic techniques are employed for localization. Moreover, VFH+ [20] is adopted to avoid a local obstacles while the robot is navigating in the rooms. Frontier-based exploration strategy [22] and Gmapping [12] algorithm are used to explore unknown environment.

[2] Relevant videos are available on our website:
http://wrighteagle.org/en/demo/index.php

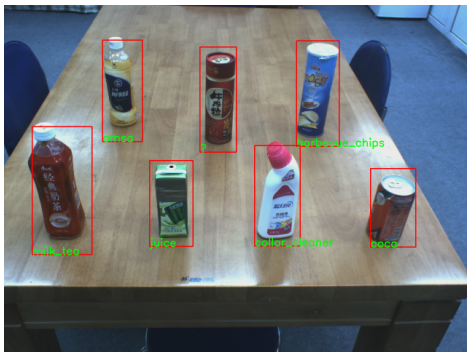

Fig. 3. Table-top object recognition results

3.2 Visual Perception

Sensors of our vision system consist of Microsoft Kinect and a high-resolution 1394 RGB camera from PointGrey. With the pre-calibrated intrinsic and extrinsic camera parameters, we obtain an aligned RGB-D image by combining the RGB image from 1394 camera with the depth image from Kinect. With such aligned RGB-D image, our vision module is capable of people awareness and object recognization and localization.

People Awareness The aligned RGB-D image is transformed into the robot's coordinate using ROS *tf* API. Since human will occupy a continuous and almost fixed-size space, we segment the point cloud into multiple connected-components, and analyze the shape of each component. Each candidate is then passed into a pre-trained HOD [18] upper body detector to decide whether it is human or not. Then a HAAR [21] face detector from *OpenCV* [4] is used to find and localize human face. If present, the VeriLook SDK will be used to identify whether it is known via face recognition.

Object Recognition We follow the approach as proposed in [17] to detect and localize table-top objects including bottles, cups, etc. The depth image is first transformed and segmented, then the largest horizontal plane is extracted using Point Cloud Library (PCL) [16], and point clouds above it are clustered into different pieces. After that the SURF feature matching against the stored features are applied to each piece [2]. The one with highest match above certain threshold is considered as a recognition. At last, to further enhance the detection performance and decrease FP rate, we check each recognized cluster and filter out those vary too much in size. Detection result is shown in Fig. 3.

3.3 Manipulation

We simplified the algorithm described in [14] by tracking a set of marks attached to arm mechanism, rather than the articulated point cloud model of the arm,

to perform online hand-eye calibration and coordination. The online calibration error of the vision-manipulator system can be less than 5 mm while the arm stops moving, which greatly improves the success ratio of manipulation.

4 Dialogue Understanding

The Human-Robot Dialogue module provides the interface for communication between users and the robot. The Speech Application Programming Interface (SAPI) developed by Microsoft is used for speech recognition and synthesis. Once a user's utterance is captured by the recognizer, it is converted into a sequence of words. The embedded dialogue manager then classifies the dialogue contribution of the input utterance by keeping track of the dialogue moves of the user. At present, the structure of the dialogue is represented as a finite state transition network. Fig. 4 shows our implementation (i.e., finite state machine) of managing a simple human-robot dialogue in which the user tells the robot facts that he/she has observed or tasks, and the robot asks for more information if needed.

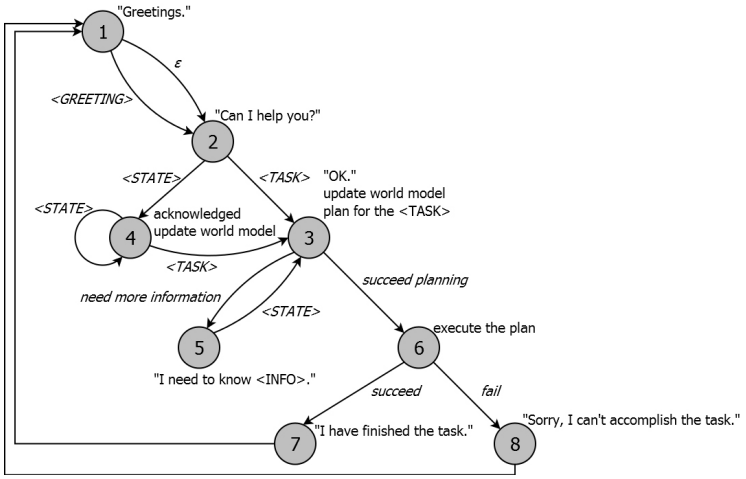

Fig. 4. The finite state machine for a simple human-robot dialogue

After the dialogue move recognition, the speech is passed into the Speech Understanding module for the translation to its semantic representation (in the form of ASP language). With the dialogue move and the semantic representation of the speech, the Speech Understanding module decides to update the World Model, which contains the information from the perceptual model and of the robot's internal state, and/or to invoke the Task Planning module for fulfilling a task.

The translation from speech to semantic representation consists of the syntactic parsing and the semantic interpretation. In the syntactic parsing, the Stanford parser [13] is employed to obtain the syntax tree of the speech. The semantic interpretation using λ-calculus [3] is then applied on the syntax tree to construct the semantics. Fig. 5 shows an example of semantic interpretation.

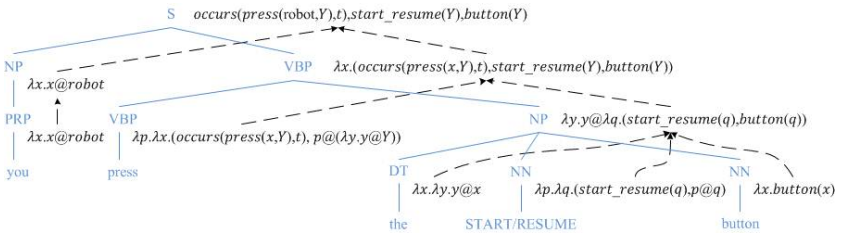

Fig. 5. An example of semantic interpretation

5 Hierarchical Task Planning

In KeJia's task planning module, a planning problem is described as an ASP program, and an ASP solver is employed to get its answer sets. Each answer set is corresponding to a high-level plan for the problem. So far, ASP solvers are not efficient enough so that the performance of task planning is poor for large scale problems. However, in the indoor service domains, a typical task (e.g., "*clean the house*") usually contains extensive steps. For instance, we have tested a 47-steps problem, and it took 25 hours to get a solution. Therefore, the efficiency of task planning is a chanllenge for KeJia.

Fortunately, there are other opportunities for speeding up resolutions with current ASP solvers as well as improving the solvers themselves. We proposed an approach to shorten the length of a plan so that the time could be saved greatly [9]. Specifically, macro-actions are employed to represent a sequence of primitive actions of the domain. In the planning procedure a macro-action acts just as a primitive action. While the adapted problem is solved, a plan including macro-actions is generated. Then all macro-actions are refined to primitive actions. At this point, we define two types of macro-actions. The first one is the *Relevant Object Macros* (ROMs), where a predefined sequence of primitive actions is used to accomplish a sub-task or to handle a certain object with multiple primitive actions sequently. The second one consists of those macro-actions learned from small-size problems of the same domain. Some macro-actions can be refined straightforwardly, that is, replaced by the corresponding primitive action sequences. But the replacement may be difficult or even impossible in some cases. A more general way is to take the refinement of a macro-action as an induced, new planning problem. In the new problem, the initial state is the

state before the macro-action's execution, the goal state is the state after its execution, and the actions are all primitive.

With the hierarchical planning method, KeJia completes task planning much more efficiently. For example, for the problem which has a 47 steps optimal plan mentioned above, KeJia got a 48 steps plan in 40 seconds with the method.

6 Conclusion

In order to meet requirements addressed in Section 1, we are developing and integrateing techniques for natural language understanding, hierarchical task planning, and knowledge acquisition. We are also developing low-leve functions that are necessary for implementing an intelligent service robot, including self-localization and navigation, visual perception, and manipulation. In order to test these techniques and the entire system, we have conducted a series of case studies involving general purpose service with incomplete or erroneous information, acquiring and reasoning with causal knowledge, learning operatations on a microwave oven through reading the manual, and acquiring open knowledge from spoken dialogue and from knowledge base.

Acknowledgement

This work is supported by the National Hi-Tech Project of China under grant 2008AA01Z150, the Natural Science Foundations of China under grant 60745002, 61175057, and USTC 985 project. Other team members besides the authors are: Xiang Ke, Zhiqiang Sui, Zhang Liu, Zhiqiang Lin, Zhe Zhao, Bin Cheng, Dongcai Lu, Keke Tang, Yingfeng Chen, and Zongjun Xie.

References

1. H. Asoh, Y. Motomura, F. Asano, I. Hara, S. Hayamizu, K. Itou, T. Kurita, T. Matsui, N. Vlassis, R. Bunschoten, et al. Jijo-2: An office robot that communicates and learns. *Intelligent Systems, IEEE*, 16(5):46–55, 2005.
2. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.
3. P. Blackburn and J. Bos. *Representation and inference for natural language: A first course in computational semantics*. CSLI Publications, Chicago, USA, 2005.
4. G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
5. W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2):3–55, 1999.
6. R. Cantrell, K. Talamadupula, P. Schermerhorn, J. Benton, S. Kambhampati, and M. Scheutz. Tell me when and why to do it!: Run-time planner model updates via natural language instruction. In *Proceedings of the 7th ACM/IEEE International Conference on Human-Robot Interaction*, pages 471–478, Boston, USA, 2012. ACM.

7. X. Chen, J. Ji, J. Jiang, G. Jin, F. Wang, and J. Xie. Developing high-level cognitive functions for service robots. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 989–996, 2010.
8. X. Chen, J. Jiang, J. Ji, G. Jin, and F. Wang. Integrating nlp with reasoning about actions for autonomous agents communicating with humans. In *Proceedings of the 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 137–140, 2009.
9. X. Chen, G. Jin, and F. Yang. Correct reasoning. chapter Extending action language C+ by formalizing composite actions, pages 134–148. Springer-Verlag, Berlin, Heidelberg, 2012.
10. X. Chen, J. Xie, J. Ji, and Z. Sui. Toward open knowledge enabling for human-robot interaction. *Journal of Human-Robot Interaction*, 1(2):100–117, 2012.
11. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *ICLP/SLP*, pages 1070–1080, 1988.
12. G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.
13. D. Klein and C. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL-03)*, pages 423–430, Sapporo Convention Center, Sapporo, Japan, 2003. ACL.
14. M. Krainin, P. Henry, X. Ren, and D. Fox. Manipulator and object tracking for in hand model acquisition. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation: Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.
15. S. Rosenthal, M. Veloso, and A. Dey. Learning accuracy and availability of humans who help mobile robots. In *Proceedings of the 25th Conference on Artificial Intelligence*, pages 60–74, San Francisco, California, USA, 2011. AAAI Press.
16. R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2011.
17. R. B. Rusu, A. Holzbach, G. Bradski, and M. Beetz. Detecting and segmenting objects for mobile manipulation. In *Proceedings of the 12th IEEE International Conference on Computer Vision: Workshop on Search in 3D and Video*, 2009.
18. L. Spinello and K. O. Arras. People detection in RGB-D data. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3838–3843. IEEE, 2011.
19. M. Tenorth and M. Beetz. KnowRob-knowledge processing for autonomous personal robots. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4261–4266, St. Louis, MO, USA, 2009. IEEE.
20. I. Ulrich and J. Borenstein. Vfh+: Reliable obstacle avoidance for fast mobile robots. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 1572–1577, 1998.
21. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.
22. B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151, 1997.