Technical Report, USTC-CS-MAS-2011-H1, Feb.15, 2011.

KeJia Project: Towards Integrated Intelligence for Service Robots

Xiaoping Chen, Guoqiang Jin, Jianmin Ji, Feng Wang and Jiongkun Xie

Multi-Agent Systems Lab., Department of Computer Science and Technology,
University of Science and Technology of China, HeFei, 230027, China
xpchen@ustc.edu.cn
http://wrighteagle.org/en/robocup/atHome

Abstract. This paper reports some progress on the project KeJia, a long-term effort aiming at integrated intelligence for service robots. We focus in this paper on the high-level cognitive functions developed on our robot KeJia, including human-robot dialogue understanding, task planning, and automatic transformation of the knowledge and information extracted from human-robot dialogues and other sources in verbal languages into the planner. We have been testing these techniques and the integrated system in RoboCup@home league competitions, as well as other case-studies involving complex tasks, general purpose service with incomplete or erroneous information, acquiring and reasoning with causal knowledge, and learning to operate a microwave oven through reading the manual.

1 Introduction

Researchers from Artificial Intelligence (AI), Robotics and related areas have shown increasing interest in developing intelligent service robots [1, 2, 5, 7, 16, 19] A service robot is generally regarded as a robot servant providing services for untrained and non-technical users in ordinary environments such as home, office, and hospital. For this purpose, an intelligent service robot must meet some requirements.

The motivation behind the work reported here is to try to develop intelligent service robots that meet following three challenging requirements. Firstly, an intelligent service robot should be able to communicate with humans naturally [1, 10, 5, 8]; in particular, it should be able to understand its users' service requests and other messages expressed in some verbal languages. Although other ways of human-robot interaction such as gesture recognition are also needed, we take spoken dialogues in verbal languages as the major means to human-robot communications. Secondly, an intelligent service robot should possess some degree of autonomy; in particular, it should be able to carry out task planning autonomously. Thirdly, an intelligent service robot should be able to learn from its experience and thus reach higher performance; in particular, we hope the robot can acquire general knowledge from human users through spoken dialogue and other sources such as the web. To our best knowledge, there is very little work on this particular requirement, while there are lots on robot learning, in which a human teaches a robot how to perform a specific task through a combination of spoken commands, observation and imitation of the human's performing that task [18].

In a long-term project aiming at the three requirements [5, 6], some general-purpose mechanisms for processing limited segments of natural languages (LSNLs), task planning, and declarative knowledge acquisition are developed and implemented on a real robot KeJia. We have tested these techniques and the whole system in RoboCup@home league competitions in past two years [4, 3] as well as other case-studies. In this paper, which also serves as the team description paper of WrightEalge for RoboCup@home 2011, we concern ourselves with some progress during the last year.

Section 2 gives an overview of KeJia system. Section 3 describes a key module of KeJia system, Pragmatic Transformation. Section 4 presents hierarchical task planning. Section 5 describes briefly the low-level functions implemented in the project. Section 6 reports some experiments on KeJia. Conclusions are given in Section 7.

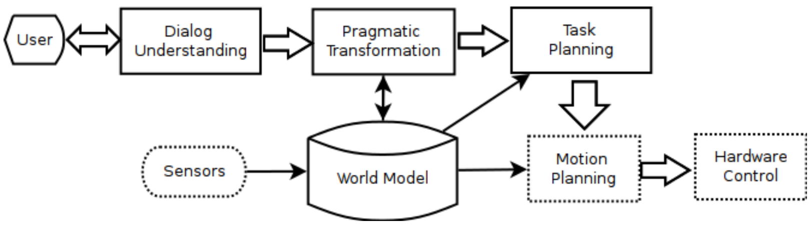
2 The Realization of Real Robot KeJia

We have made a new robot, KeJia-2, as shown in Figure 1a. Its hardware components include a laser range finder, a stereo camera, and two arms for manipulating portable items, all installed on an omnidirectional wheeled base. The computational resources consist of two on-board 4-core PCs. Neither additional computational resources off-board nor remote control is needed for the robot when it performs its tasks.

The flowchart of KeJia system is shown in Figure 1b. The robot is driven by input from human-robot dialogue. The spoken dialogue between robot KeJia and its users is restricted to some limited segments of natural languages (LSNLs). A specific LSNL is defined with a fixed vocabulary and a simplified syntax, a subset of the syntax of some natural language. Service queries, descriptions about the states of the environment, knowledge of the world, instructions about new tasks and so on can be expressed in these LSNLs. The expressive power of an LSNL is mainly determined by its scale.

The texts drawn by standard speech recognition software from the spoken dialogue is processed syntactically with the Stanford parser [13], and then semantically through a lazy semantic interpreter we developed in KeJia project [5]. The results of the semantic analysis are represented in a form similar to the Discourse Representation Structure (DRS) [12], with an extended semantics. The information in this internal representation is transformed by the pragmatic transformation module (see Section 3 for more details) into the Task Planning module, which is based on a nonmonotonic logic language, Answer Set Programming (ASP) [9], where the information and knowledge are represented as an ASP program, and some ASP solver is employed to generate a course of actions for the user's task. The Task Planner will re-plan during the execution of a high-

(a) KeJia-2



(b) The flowchart of KeJia's high-level functions

Fig. 1: Hardware and Software of KeJia

level plan when necessary, according to the up-to-the-minute request of the user and/or the timely state of the World Model (e.g., when there appears any new request, some executive failure or abnormity).

A high-level plan generated by the Task Planner is fed into the Motion Planner. Each action is designed as a primitive for KeJia's task planning and can be realized by KeJia's motion planner. For each primitive action in a high-level plan, the Motion Planner will try to make a low-level plan, and the Robot Controller will execute the low-level plan autonomously. The execution causes some changes of the environment and the state of the robot itself. The World Model is updated accordingly with the information perceived by the sensors.

The Motion Planner 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. As a consequence, an intelligent service robot with rich high-level cognitive functions could not be captured by a single formal model. We draw the line between task and motion planning.

3 The Pragmatic Transformation

The Pragmatic Transformation is one of KeJia's key components, with which there is very little work in literature. Its input is a set of LSNL-sentences in some internal representation, expressing the user's requests (task) and other messages (such as descriptions about the states of the environment, knowledge of the world). Its output is an ASP program which contains all the information for solving the user task. The pragmatic transformation is completed by executing a set of transformation rules, which are generally built up on top of another kind of more elementary rules, called interpretation rules.

The interpretation is not a linguistic issue, but the symbolic grounding problem [11], i.e., how to link the abstract concepts (expressed with the words from LSNLs in KeJia system) to the perception and actuation of the robot. This problem is of essential importance for knowledge representation of autonomous robots [19].

1. Interpretation Rules. Here we only describe three main sorts of primitive interpretation rules informally. The first sort is for nouns, pronouns and adjectives in LSNLs, which are linked directly to the World Model and thus mapped into the low-level abstraction of the robots perceptual data. In particular, a noun or pronoun is interpreted as an object, a set of objects, or an attribute of the World Model, while an adjective an attribute of the World Model. For instance, noun "can" is interpreted as the set of cans appeared in the World Model and adjective "red" is linked to the corresponding value of the World Model's feature "color". The semantic relationships between these words and their referents are clear and conform to our intuitive understanding of these relationships.

The second sort of primitive interpretation rules is for verbs in LSNLs. A verb is eventually mapped into a course of low-level actions of the robot (The mapping depends on the context where the verb occurs, but we omit this contextual issue here). Generally, the mapping of some verbs may be expressed only in non-linguistic ways [15], and thus may not be formalized completely in logic. On the other hand, most verbs should not be mapped directly into the robots routines; otherwise, there would be no task and/or motion planning for the realization of the actions that these verbs refer to. To support the specification of verbs mapping, we introduced a substrate symbolic language, which is also available for other purposes. This language consists of primitive actions (as mentioned in Section 2) and other built-in identifiers, which are defined and realized in terms of the robots routines and parameters. The substrate symbolic language is designed so that all the verbs in LSNLs can be defined in terms of it and further realized through the robots routines and parameters. Examples will be given below where transformation rules are presented.

The third sort of primitive interpretation rules is for words and linguistic constructors corresponding to logical operators. So far in KeJia Project, we have considered two words corresponding to two most important logical operators, "if" (including "if-then") and "not" [5]. These two words can have diverse functions. In everyday dialogue, however, they mainly function as commonsense terms. There are three cases where "not" appears in current LSNL-sentences. (i) "not" is used to form a negative, imperative sentence, such as "do not open the door". The whole sentence expresses that some action is forbidden, which can be translated naturally into an ASP constraint. It seems to us that this is the best way of handling this case. (ii) "not" modifies the main verb of a sentence or clause. These sentences or clauses should be handled similarly to a negative, imperative sentence as above. In particular, negative if-clauses should be understood as defeasible conditions and thus rewritten as ASP rules. (iii) "not" modifies "anything", representing "nothing". The sentence or clause should be translated into an ASP rule too. In all the cases, word "not" is translated naturally or approximately into the negation-as-failure operator, *not.* It turns out that nonmonotonic logic plays an essential role in these cases, making the interpretations extremely simple and efficient.

2. Transformation rules. For sake of simplicity, we omit the first-order format of DRS here and describe the transformation rules as mappings from LSNL-sentences into ASP-rules. KeJia's semantic analyzer classifies the LSNL-sentences into three types, for each of which there is a set of corresponding transformation rules.

(a) LSNL-sentences that just provide information about the environment will be transformed into ASP facts and/or rules. For instance, "The book is on the table." is converted to the following ASP rules:

$$holds(samelocation(X,Y),0) \leftarrow book(X), table(Y).$$

where "samelocation" is a built-in identifier of the substrate symbolic language, which is interpreted by pre-defined parameters, such as some gridding of the environment. Therefore, the transformation incorporates the effect of some interpretation rules. With this rule, when needed, the robot will search the book at the same location of the table according to the parameters, by executing the corresponding routine. This type of transformation is rather simple. Also the adding of the knowledge and the utilization of it are easy in logic formalisms.

(b) A LSNL-sentence representing a simple task will be transformed into an ASP goal. For instance, "Give Jim a red bottle" is a simple task and will be transformed as following ASP-rules:

$$\begin{aligned} goal \leftarrow & holds(samelocation(X,Y), lasttime), holds(handempty, lasttime), \\ & Jim(X), red(Y), bottle(Y). \\ \leftarrow & not\, goal. \end{aligned}$$

The first rule above states that, a specifically introduced goal will be reached if the task is accomplished, while the content of the goal is that the red bottle and Jim are at the same location and the hand of the robot is empty. The second rule specifies that the goal must be achieved. "handempty" is another built-in identifier, interpreted by robot routine. Eventually, "Give Jim a red bottle" is realized as "reach a location close enough to Jim, and put down the red bottle at the location or hand it over to Jim". Obviously, this low-level realization of the task is very hard to be captured in a logical formalism. Hence it is crucial to appropriately separate low-level realization from high-level planning; otherwise, nonmonotonic logics could not be available for the high-level functions of intelligent robots.

(c) Causal knowledge obtained from spoken dialogue between the robot and its users are also transformed into ASP rules. Consider the sentence: "the object will fall, if the object is on the sticking-out end of the board and there is nothing on the other end of the board." This sentence expresses a piece of knowledge regarding a special form of notion of balance. Based on relevant interpretation rules, this is translated to the following ASP rule:

$$\begin{aligned} \mathit{holds}(\mathit{falling}(X),T) \leftarrow & \mathit{holds}(\mathit{on}(X,Y),T), \mathit{sticking\_out}(Y), \mathit{endof}(Y,Z), \\ & \mathit{board}(Z), \mathit{endof}(U,Z), \mathit{not}\, \mathit{holds}(\mathit{on}(V,U),T). \end{aligned}$$

where sticking_out, endof and board are built-in identifiers and can be handled by the Motion Planner and the robot's perceptual system. Once this rule is added into the Task Planner, robot KeJia can make use of it in task planning autonomously [5].

According to the interpretation and transformation rules, other kinds of domain knowledge can be converted and added into ASP program similarly. For example, the description of primitive action *pick up* is given in KeJia's Task Planner as follows:

$$\begin{aligned}
\mathit{occurs}(\mathit{pick\_up}(A),T) \leftarrow & \mathit{not}\, \neg \mathit{occurs}(\mathit{pick\_up}(A),T). \\
\neg \mathit{occurs}(\mathit{pick\_up}(A),T) \leftarrow & \mathit{not}\, \mathit{occurs}(\mathit{pick\_up}(A),T). \\
\neg \mathit{occurs}(\mathit{pick\_up}(A),T) \leftarrow & \mathit{not}\, \mathit{holds}(\mathit{samelocation}(\mathit{agent},A),T). \\
\neg \mathit{occurs}(\mathit{pick\_up}(A),T) \leftarrow & \mathit{not}\, \mathit{holds}(\mathit{handempty},T). \\
\mathit{holds}(\mathit{holding}(A),T+1) \leftarrow & \mathit{occurs}(\mathit{pick\_up}(A),T), \mathit{size}(A,\mathit{small}), T < \mathit{lasttime}. \\
\neg \mathit{holds}(\mathit{handempty},T) \leftarrow & \mathit{holds}(\mathit{holding}(A),T). \\
\mathit{holds}(\mathit{holding}(A),T+1) \leftarrow & \mathit{holds}(\mathit{holding}(A),T), \mathit{not}\, \neg \mathit{holds}(\mathit{holding}(A),T+1), \\
& T < \mathit{lasttime}. \\
\neg \mathit{holds}(\mathit{holding}(A),T+1) \leftarrow & \neg \mathit{holds}(\mathit{holding}(A),T), \mathit{not}\, \mathit{holds}(\mathit{holding}(A),T+1), \\
& T < \mathit{lasttime}. \\
\mathit{holds}(\mathit{handempty},T+1) \leftarrow & \mathit{holds}(\mathit{handempty},T), \mathit{not}\, \neg \mathit{holds}(\mathit{handempty},T+1), \\
& T < \mathit{lasttime}. \\
\neg \mathit{holds}(\mathit{handempty},T+1) \leftarrow & \neg \mathit{holds}(\mathit{handempty},T), \mathit{not}\, \mathit{holds}(\mathit{handempty},T+1), \\
& T < \mathit{lasttime}.
\end{aligned}$$

One of our long-term goals is to teach KeJia general knowledge, including action descriptions, through spoken dialogues in LSNLs. But now this is not always feasible. The first reason is that the current interpretation and transformation rules are not complete or powerful enough. The second one is the efficiency of current ASP solvers. We have to do some manual optimization on the ASP programs.

4 Hierarchical Task Planning

In KeJia's task planner, a planning problem is described as an ASP program, and then an ASP solver is called to get the answer sets of the program, each corresponding to a high-level plan of the problem. For small problems that have a plan with less than 20 steps, this works well. But this is not the case when the problem is large, since the current ASP solvers are not efficient enough. In domestic domains, a typical task such as "clean the house" may contain much more steps. We have tested a problem which has an optimal plan consisting of 47 steps. It ,took 25 hours to get a single solution.

Along with the development of ASP solvers, the ASP planning technique is hopeful to be able to handle larger and larger problems in the future. Meanwhile, there are other opportunities of speeding up solutions with the current ASP solvers. In planning, the longer a plan is, the more time will be spent on one step forwarding. So it is not surprising that a 20 steps plan takes twice the time as a 19 steps plan for the same problem. Thus if we can shorten the plan length, the time for solving the problem can be greatly saved. One of the promising techniques is to use macro-actions in the planning.

A macro-action represents a sequence of primitive actions of the domain. In planning procedure a macro-action acts just as a primitive action, when it is added to the original domains. In solving the adapted problem that results from the extension of the macro-actions, plans that may include macro-actions are generated. Then for any of these plans which is chosen for further processing, all the macro-actions in it are refined to primitive actions and thus a plan without macro-actions is obtained as the solution to the original planning problem.

Currently we are using two types of macro-actions in KeJia's system. First one is the "Relevant Object Macros (ROMs)", where a pre-defined 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-step optimal plan mentioned above, KeJia got a 48-step plan in 40 seconds with the method.

5 Low-level Functions

Low-level functions are necessary and crucial in some cases for the development of an intelligent service robot. Here we describe briefly navigation, perception, and manipulation implemented on our robot KeJia.

Navigation A 2D occupancy grid map is learned from laser scans, collected by the robot though a round travel within the rooms aforehand [10]. The map is then manually annotated with the approximate location and/or area of rooms, doors, furniture and other interested objects. And thus a topological map can be automatically generated, which will be used by the global path planner and imported as a part of prior world model. Scanning match and probabilistic techniques are employed for localization, and VFH+ [20] is adopted to avoid a local obstacles while the robot is navigating in the rooms.

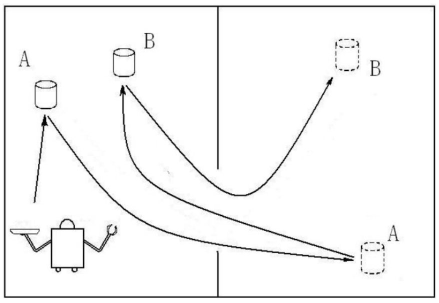
Perception We follow the approach proposed in [17] to detect and locate the tabletop objects, such as bottles, cups, appliances, etc. A further effort was taken to make the robot be able to carry out challenging manipulation task e.g. operating the household appliance. Take the microwave oven for example, to preciously estimate the 6-DOF pose of the oven's body, buttons and even the opening angle of the oven's door, our current implementation employs a model-based method, which aligns the 3D point cloud from stereo vision with the given geometric model of the oven and achieves a repeatable accuracy of less than 1 mm. Meanwhile , the high-resolution image captured by another camera is analyzed with a template-based method to recognize the numbers and firepower displayed on the oven's control panel.

Manipulation Robotic arm tracking technique described in [14] is substantially useful to accomplish high-precision manipulation tasks, especially for the low-cost arm with uncertain clearance in the joints. We simplified the algorithm 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.

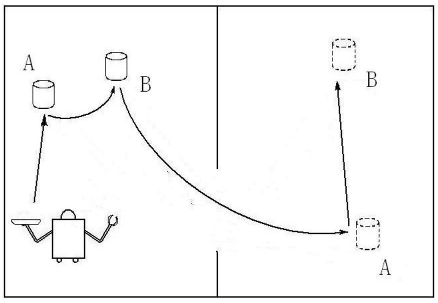
6 Experiments

We have conducted a series of case studies in KeJia project. In the first class of case-study, we took standard tasks in RoboCup@home competitions as benchmarks, e.g. building a map of an unknown environment, identifying humans, following an unknown person through a dynamical environment. The second class of case study has been carried out where KeJia was given some problems, each specifically designed for testing a certain aspect of the whole system. In particular, some of the tasks can only be accomplished by virtue of acquiring new knowledge from spoken dialogues and/or other sources in natural languages and making use of it in task planning for the problems. Some of the case studies that aim to examine the high-level cognitive functions are described briefly below.

Planning for Complex Tasks: A task is complex, if it consists of multiple component tasks/goals. An additional factor of difficulty is introduced by the existence of related goals in a complex task. Figure 3 shows an example. Suppose you and your friend are setting in the living room and you ask your robot to fetch two cans of beer from the dining room for you and your friend. This request is a complex task consisting of two related goals, moving the first can from the dining room to the living room and moving the second from the dining room to the living room. If the robot cannot understand or make use of the relatedness of the goals, it will fetch the cans one by one separately, as shown in Figure 3(a). Obviously, this is not necessarily optimal and is typically inefficient. An optimal plan is shown in Figure 3(b). This is the way our robot KeJia took[1]. This requires that the robot understands and makes use of the relationships among the actions needed for reaching the related goals. A further requirement can be identified.



(a) Complete component tasks separately



(b) Complete component tasks interleavingly

Fig. 2: Complex Task Plans

In many cases, a user can only specify the goal of a task, without knowing how to reach it. For example, you may ask a robot to prepare some kind of special food for you, but do not know how to prepare it. In this situation, the robot has to work out by itself a course of actions whose execution will reach the goal. The case-study shows that KeJia possesses this ability, at least in its task planning, to some extent.

General Purpose Service with Incomplete or Erroneous Information: The General Purpose Service has been taken as a standard test in RoboCup@home 2010 competitions. In this test[2], a robot is given multiple tasks to complete. This is a generalized form of complex tasks, in which the robot may be requested to perform a complex task that is underspecified or even specified with erroneous information. These possibilities require that the robot must be able to understand these kinds of information and handle the situations properly.

We have conducted a series of experiments on different tasks of general purpose service. In one of the experiments, the user firstly gave an underspecified instruction to KeJia. KeJia discovered this underspecification and actively requested for more information to find out the "true" intention of the user and then completed the task accordingly. In addition, the user asked KeJia to perform an unachievable task. KeJia found the infeasibility of the task during the execution of her tentative plan and reported the infeasibility to the user, instead of blind execution of users commands regardless of any potential problems. The experiments show that KeJia possesses the ability of handling the incomplete or erroneous information to some extent.

[1] The Video can be found at http://wrighteagle.org/en/demo/ComplexTasks.php
[2] http://wrighteagle.org/en/robocup/atHome/video/WEHome2011Quali.wmv

Acquiring from Spoken Dialogue and Reasoning with Causal Knowledge: In this demo[3], KeJia was taught some causal knowledge through spoken dialogue. Using the acquired knowledge, the robot generated a plan that cannot be generated with the knowledge and complete the task. In the experiments, we took a version of KeJia without any build-in knowledge about "balance", "fall", or any other equivalents. KeJia was told in the human-robot spoken dialogue that an object will fall if it is on the sticking-out end of a board and there is nothing on the other end of the board. With the knowledge, KeJia accomplished the task of moving the green can while avoiding anything falling with the following plan: moving the red can and put it on the table first, and then picking up the green can. This indicates that KeJia's ability is substantially raised by knowledge acquisition through spoken dialogue. It took no more than 0.5 second for KeJia to "understand" the knowledge, transform it into some internal representation, and generate the plan with the taught knowledge.

Learn to Operate a Microwave Oven through Reading the Manual: This demo[4] was repeated dozens of times to the public in the 12th China Hitech Fair (Shenzhen, Nov. 16-21, 2010). During the demonstration, the user gave oral commands in Chinese to robot KeJia, requesting her to prepare breakfast, in particular, to heat up milk and/or bread. In the beginning, KeJia had no complete knowledge about the microwave oven; for instance, she did not know the functions of the buttons on the control panel. While the user suggested her "reading the manual", KeJia immediately browsed and downloaded the manual from the internet, read the manual in Chinese and learnt the relevant knowledge in it. With the acquired knowledge, KeJia managed to plan a course of actions and achieve the entire task by using the microwave oven to heat up food. The actions she executed included opening the microwave oven door, putting the food into the microwave oven, closing the door, pushing button "Start/Re-heat" to start heating, pushing button "Pause/Cancel" to stop heating, opening the oven and finally bringing the heated food to the user. In this process, KeJia checked her actions of pushing buttons by monitoring the readings on the digital screen, and took remedial actions when necessary.

This demo reflects a novel and more challenging test on the architecture and core technologies of KeJia. For this test, we slightly extended KeJia's knowledge base of languages, while substantially augmented its manipulation abilities (i.e., opening and closing the door of microwave ovens, pushing buttons) and perceptual abilities (i.e., detecting a microwave oven and recognizing the readings on digital screen), as described briefly in Section 5. The results indicate that based on the architecture and technologies we developed, hopefully in the future, intelligent robots would be enabled to autonomously acquire knowledge expressed in natural languages and thereby incrementally improve their performance on general-purpose services.

[3] http://ai.ustc.edu.cn/en/demo

[4] http://www.wrighteagle.org/en/demo/microwaveoven.php

7 Conclusions

In this paper, we report some progress in KeJia project and mainly concern ourselves with the high-level cognitive functions of the robots. We are developing and integrating techniques for natural language understanding for limited fragments of English and China, hierarchical task planning, and automatic transformation of the knowledge and information drawn from human-robot dialogue and web pages into some form available by the task planner. We are also developing Low-level functions that are necessary for implementing an intelligent service robot, including navigation, perception, and manipulation. This effort aims to meet the requirements mentioned in Section 1. In order to test these techniques and the entire system, we have conducted a series of case studies involving complex tasks, general purpose service with incomplete or erroneous information, acquiring and reasoning with causal knowledge, and learning to operate a microwave oven through reading the manual.

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, and USTC 985 project. Other team members besides the authors are: Kai Chen, Min Cheng, Xiang Ke, Zhiqiang Sui. We are also grateful to Michael Gelfond, Fangzhen Lin, Peter Stone, Mary-Anne Williams, and Shlomo Zilberstein for valuable discussions about the project. A special thank is given to Benjamin Johnston for his detailed helps.

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. 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.
3. X. Chen, J. Ji, J. Jiang, and G. Jin. WrightEagle Team Description for RoboCup@ Home 2009. Technical report, Technical report, Department of Computer Science and Technology, University of Science and Technology of China, 2009.
4. X. Chen, J. Ji, J. Jiang, and G. Jin. Progress of Ke Jia Project. Technical report, Technical report, Department of Computer Science and Technology, University of Science and Technology of China, 2010.
5. 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 Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-10)*, pages 989–996, 2010.

6. 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 (IAT-09)*, pages 137–140, 2009.
7. A. Ferrein and G. Lakemeyer. Logic-based robot control in highly dynamic domains. *Robotics and Autonomous Systems*, 56(11):980–991, 2008.
8. T. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots. *Robotics and autonomous systems*, 42(3-4):143–166, 2003.
9. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *ICLP/SLP*, pages 1070–1080, 1988.
10. G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *Robotics, IEEE Transactions on*, 23(1):34–46, 2007.
11. S. Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, 1990.
12. H. Kamp and U. Reyle. From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory. *Computational Linguistics*, 21(2):265–268.
13. D. Klein and C. Manning. Fast exact inference with a factored model for natural language parsing. *Advances in neural information processing systems*, pages 3–10, 2003.
14. M. Krainin, P. Henry, X. Ren, and D. Fox. Manipulator and object tracking for in hand model acquisition. In *Proc. of the Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation at the Int. Conf. on Robotics & Automation (ICRA), Anchorage, Alaska*, 2010.
15. J. McCarthy. Elaboration tolerance. In *Common Sense*, volume 98. Citeseer, 1998.
16. M. Quigley, E. Berger, A. Ng, et al. Stair: Hardware and software architecture. In *AAAI 2007 Robotics Workshop, Vancouver, BC*, 2007.
17. R. Rusu, A. Holzbach, M. Beetz, and G. Bradski. Detecting and segmenting objects for mobile manipulation. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 47–54. IEEE, 2010.
18. P. Rybski, K. Yoon, J. Stolarz, and M. Veloso. Interactive robot task training through dialog and demonstration. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction(HRI-07)*, pages 49–56. ACM, 2007.
19. M. Tenorth and M. Beetz. KnowRobknowledge processing for autonomous personal robots. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4261–4266. IEEE, 2009.
20. I. Ulrich and J. Borenstein. VFH+: Reliable obstacle avoidance for fast mobile robots. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1572–1577. IEEE, 2002.