This is the text version of the plenary talk with the same title at *The Third International Conference on Robot Intelligence Technology and Applications*, Beijing, Nov. 7-8, 2014, which will also appear in the proceedings of the conference.

Expanding Robot Intelligence with Open Knowledge under the Context of Human-Robot Society

Xiaoping Chen
Computer School
University of Science and Technology of China
Hefei 230026, China
xpchen@ustc.edu.cn

Expanding Robot Intelligence with Open Knowledge under the Context of Human-Robot Society

Xiaoping Chen
Computer School
University of Science and Technology of China
Hefei 230026, China
xpchen@ustc.edu.cn

Abstract. It is desirable to develop intelligent service robots capable of working in large domains. These robots are required to be able to perform user tasks that are not preprogrammed and that are expressed naturally. This paper makes an analysis of infeasibility of modeling such a robot in specified representations, which was assumed in traditional approaches, and proposes the thesis of underspecification that underspecified representations should be employed to model these robots. The feasibility of this open knowledge approach is preliminarily demonstrated by an implemented robot prototype, OK-KeJia, and experiments on large test sets with it.

1 Introduction

The world is approaching a human-robot society where humans and robots cooperate with each other in various environments such as offices, homes, and hospitals. Under this context, the robots are required to be able to perform user tasks that have not been preprogrammed [1], [2], [3]. It is also desirable that the robots can understand naturalistic expressions of user tasks [4], [5], [6]. These requirements demand intelligent robots substantially on their capability of working in large domains. Knowledge plays an essential role in satisfying either of the above challenging requirements. As more and more on-line knowledge resources become available (including ontologies and large-scale knowledge bases such as Cyc and the Open Mind Family), there are new opportunities to make use of open knowledge, i.e., the knowledge from open-source knowledge resources to expand robot intelligence [1], [2], [3], [6]. The idea is to develop robots that can make use of open knowledge, as well as perceived information of the task setting, to meet a user request at hand. Superficially, this approach seems to be a traditional one, which was popular in the past. But it is not the case.

A distinguishing feature of the approach is that it is based on underspecification of open knowledge and user requests [2], while traditional approaches were typically based on an assumption that all pieces of knowledge and user requests are fully specified. For example, a user request "help me carry this box" is not fully specified in the sense that the request itself does not provide complete definitions of its components such as "carry". If a robot has the action capability of carrying the box but does not know that "carry" can be realized by executing its action of carrying, the robot still cannot meet the request. If a robot does not know some relationship between a user task and an action of it, such as that between "carry" and action of carrying, we say that there is a knowledge gap between the user request and the robot's local knowledge [6]. It seems that knowledge gaps can be avoided with fully specified user requests and relevant knowledge, as traditional approaches assumed. However, this assumption suffers from serious shortages and has not proved its potential for development of robots or agents applicable to large domains.

We proposed an approach for development of robots applicable to large domains by making use of open knowledge [6]. A radical challenge to this proposal lies in the fact that open knowledge and naturally expressed user requests are typically underspecified. Section 2 provides an analysis on the challenge and put forth the thesis of underspecification. Section 3 and 4 demonstrate a preliminary realization of the approach, based on the thesis.

2 Underspecification

A domain D is defined in this paper as a set of states of the real world, generally about a certain subject, where each state consists of all information of the world at a certain moment. Therefore, a domain is not a representation of some part of the world but just that part of the world itself. Consequently, a robot working in a domain D needs to have a representation of D in order to conduct cognitive activities on D, such as reasoning about, planning for and learning in the domain.

A domain representation is a pair $M = <A, K>$, where A is a set of presuppositions about D and K is a knowledge base of D. Sometimes A and K are called the domain specification and domain knowledge base of D, respectively. A domain specification A of D specifies all the conditions that determine if a state s of the world is in D; in other words, s is in D if and only if s satisfies all the conditions in A. The knowledge base of a domain includes facts and propositions that are chosen by the designer to model the domain.

We assume that any robot possesses a set of basic abilities of perception and action, and each of these basic abilities can be defined as a *primitive* in some internal representation of the robot. We also assume that any user task can be translated into some internal representation of a robot. Therefore, in order to model a robot working in a domain and trying to meet user tasks, one needs the representations of the domain, the primitives of the robot, and the user tasks. For simplification, we assume the three representations are expressed in the same representational language and hence form a uniform representation of the robot.

A representation of a robot is *specified* if it is well-defined, i.e., any component of it is defined in terms of the primitives of the robot. This implies that any user tasks can be realized through some "combination" of the robot's primitive actions and perceptions. A representation of a robot is *underspecified* if there is any component that is not completely defined, i.e., is undefined or partially defined, in terms of the primitives of the robot. In particular, if a user task is not defined in the representation of a robot, we say that there is a *knowledge gap* between the task and the representation of the robot. A domain knowledge base of a robot has a knowledge gap, when a component of the knowledge base is not defined in terms of the robot's primitives.

Traditional AI approaches typically adopted the assumption that any intelligent system can be modeled by a specified representation in some form of logic, mathematics, or at least computer program. However, AI and Robotics research practice does not provide any strong enough evidence that this assumption is validate for a large domain or a large set of naturally expressed user tasks. An excellent analysis was provided in [7] revealing why logical axiomatization representations based on this assumption cannot work for large domains. In this paper, we advance this analysis by putting forth a stronger assertion, *the thesis of underspecification*: It is infeasible to model a robot working in a large domain or interacting naturally with humans by a specified representation; instead, underspecified representations should be employed. It follows from this thesis that one needs to develop novel technology of handling underspecification of robot representations, so that the robots can make use of underspecified representations to solve complicated tasks in large domains.

Several causes give rise to inevitability of underspecified representations of large domains.

Case 1. Details of knowledge from different sources may be inconsistent, while some high-level part of the knowledge is consistent. For example, Chinese people agree on a proverb that "Good men will be rewarded for their kindness sooner or later." This piece of knowledge is underspecified since one of its components, "good men", is not defined in it. However, attaching a definition of "good men" to the proverb is problematic, since people have different definitions of "good men". For a certain person, someone thinks the person is a "good man" and someone else does not. In order to get rid of the inconsistency, no matter which representational formulation is adopted, we are forced to give up the completeness of the knowledge representation and keep as common knowledge the proverb itself, with "good men" as an undefined component (concept) of it. This is a reason why common knowledge is generally underspecified. Here the true point is not that incomplete representations of common knowledge must be avoided, but that common knowledge should be incompletely represented in order to maintain consistency of common knowledge.

Case 2. Specifiedness of a representation results in its closeness, which in turn hinders the scalability and flexibility of the representation. For example, there was a standard ranking and naming system for brothers in ancient China, where four ordinal words were used in brothers' names to indicate their orders: BO (meaning "the eldest"), ZHONG ("the second eldest"), SHU ("the second youngest"), and JI ("the youngest"). These words were tied up with a set of default rules, specifying the properties of the ranking system. If a fifth brother was born to the family, then the original ranking system should be updated to cover the knowledge about all the five brothers. But this update could not be done by just adding some new rules about the fifth brother into the ranking system, since some properties (e.g., "no brother is younger than JI") in the original system prevents this kind of update. In other words, the update will involve belief revision, i.e., cancelling some rules from the original ranking system before adding more new rules into it. In order to avoid belief revision in the update, one has to leave out some properties such as that "no brother is younger than JI" from the ranking system and hence make it underspecified as the definition of JI is incomplete, with a distinguishing feature of JI missing.

Case 3. Generally an existing domain specification is either partially and implicitly contained in the coupled domain representation or explicitly described in an informal meta-language such as natural language. In the former sub-case there are no explicitly expressed presuppositions of the domain, while in the later sub-case there is no well-defined domain specification. Therefore, existing domain representations are typically underspecified since their domain specifications are not well-defined. For example, the domain specification of the blacks world is given in some natural language (usually with figures) in text books. There is no well-formed definition with which a robot can judge if a world state is in the domain. Consequently, a robot cannot decide if the domain knowledge of the blacks world should be employed in decision-making on such a world state. This is another reason why it is very difficult for existing robots to work in large domains. Consequently, an existing robot operates in a structured or small domain with complete and consistent domain knowledge, avoiding any judgment involving a domain specification.

We emphasize that the above discussion about underspecified representations is different from previous discussions about incompleteness of knowledge. A distinction between them is that the discussion here reveals that sometimes more knowledge will cause big trouble which is very hard to overcome. Both case 1 and 2 are examples. This observation leads to the idea of making use of underspecified representations for robots that operate in large domains. This way, underspecification is not taken as a disadvantage, but a demand that must be satisfied and an advantage that should be made use of in development of intelligent robots working in large domains.

On the other hand, however, underspecified representations generate new challenges to research on robot intelligence, since they violate a basic assumption behind existing representation and reasoning technologies. For example, how to handle undefined components in a domain representation? We proposed a preliminary solution to this challenge and have made efforts toward this goal in recent years, as described in Section 3 and Section 4.

3 The Approach

Below we propose three principles, which direct our efforts on the Open Knowledge approach [6], [8]. Each of these principles reflects a turn of methodology of intelligent system development.

(1) *Turn from handcrafted knowledge to open knowledge*. Intelligent robots should make use of open knowledge from on-line resources besides a robot's local knowledge base, instead of only from the later, which is preprogrammed and generally not open. More and more open knowledge resources become available on the web, such as Cyc, the Open Mind Indoor Common Sense database [9], ontologies, digital dictionaries, and household appliances manuals. This suggests new opportunities that robots increase their performance using an increasing amount of open knowledge from the web. This implies that a robot's performance will increase as open knowledge increases, even if all the other aspects of the robot remain unchanged.

(2) *Turn from specified representations to underspecified representations*. Intelligent robots' cognition (decision-making, planning, knowledge acquisition, etc.) should be based on underspecified representations, instead of on specified representations. Most of open knowledge available now is expressed in underspecified representations. As pointed out in Section 2, this is an advantage one can make use of. For example, common knowledge is typically underspecified and necessary for robot cooperation and human-robot interaction.

(3) *Turn from ungrounding to grounding*. An intelligent robot must connect its low-level system with the world through *grounding*, instead of preprogrammed mapping between them [10], [11]. Open knowledge expressed in underspecified representations demands more on the robot's grounding mechanism, since there are gaps between some pieces of open knowledge and the real world [6]. Traditionally, these gaps were completed by preprogrammed mapping between symbols and their referents, which was only practical for small domains.

In the rest part of this paper, we focus on how to enhance robots' intelligence through making use of open knowledge in underspecified representations, based on the first two principles above. The challenge is cast as *the knowledge rehabilitation problem* and can be explained by Figure 1. Suppose a robot is given a user task for which there is a knowledge gap in the robot's local knowledge stored in *LKB*. The idea is for the robot to try to extract a set (denoted by *EK*) of pieces of knowledge from the open knowledge resources available, and employ the knowledge in *EK* and *LKB* to generate a plan of the task, so that the task can be achieved by executing the plan.

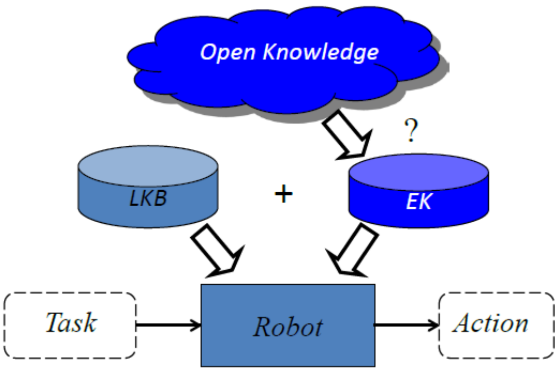

Fig. 1. The knowledge rehabilitation problem

There frequently exist multiple knowledge gaps for a single task. Therefore, the knowledge rehabilitation must be realized with an iterative procedure in which a robot tries to find a piece of external knowledge for every knowledge gap in its planning for a task [6]. Generally there is no guarantee that a task from an open-ended set can always be solved through this procedure. It is also proved that computation related to the procedure is generally intractable [12]. Fortunately, we have identified a special case which is computationally tractable and most of common open knowledge fits to this case [8]. A series of experiments have been carried out with encouraging results [6], [8].

Researchers realized very early in Knowledge Engineering (KE) that knowledge plays an essential role in development of intelligent systems and that a complete knowledge base is very hard to build. Traditionally, knowledge shortage in an intelligent system was identified and completed by handcraft. In our approach, however, it is robots that take charge of this work autonomously. But a robot in our approach does not pursue development of a complete or an increasingly complete knowledge base through its accumulation of external knowledge. Instead, the robot takes the whole body of the open knowledge available as its growing external knowledge base and relies on the whole human-robot society to build this knowledge base gradually. A robot in our approach just extracts and employs a set of knowledge pieces for a user task at hand each time.

4 The OK-KeJia Prototype and Experimental Results

We implemented a robot prototype on the overall architecture shown in Figure 2. Some learning techniques have not been implemented or integrated into the prototype. Here we briefly describe the other main modules of the prototype except the NLP module (please see [6], [13].)

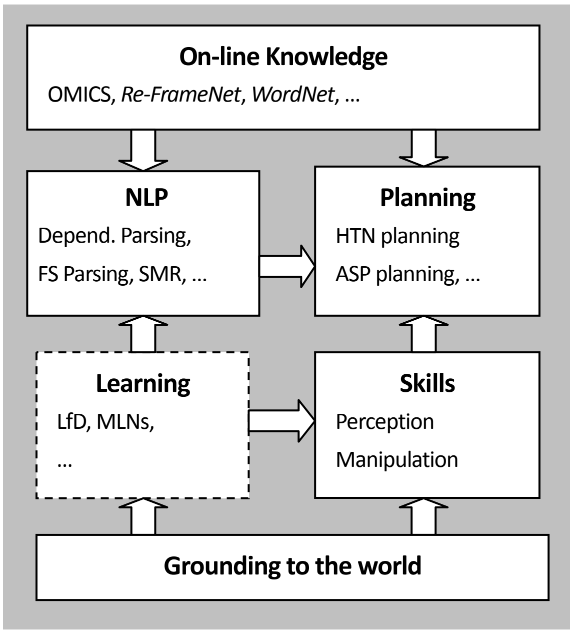

Fig. 2. Implementation architecture

4.1 Knowledge Resources and Tools

An extensive collection of knowledge for autonomous indoor robots was gathered from Internet users in the OMICS project (Gupta & Kochenderfer, 2004). The knowledge was input into sentence templates by users, censored by administrators, and then converted into and stored as tuples, of which most elements are English phrases. At present OMICS contains 48 tables capturing different sorts of knowledge, including a *Help* and a *Tasks/Steps* table. Each tuple of *Help* maps a user desire to a task that may meet it (e.g., <*feel thirsty*, *by offering drink*>). Each tuple of *Tasks/Steps* decomposes a task into steps (e.g., <*serve a drink*, 0. *get a glass*, 1. *get a bottle,* 2. *fill class from bottle,* 3. *give class to person*>). Therefore, OMICS provides a good example of hierarchism of naturalistic instructions, where a high-level user request is reduced to lower-level tasks. Another feature of OMICS is that elements of any tuple in an OMICS table are semantically related according to a predefined template. This facilitates the semantic interpretation of OMICS tuples to a large extent.

FrameNet[1] is a digital dictionary providing rich semantic information for action verbs. It groups action verbs into"Frames" and specifies word definitions in terms of semantic roles called *Frame Elements* (FEs) for each Frame [14]. We discover that these connections between an action verb and its semantic roles are very helpful for resolving underspecification of naturalistic instructions. However, the knowledge cannot be used by robots, since it is not formalized in *FrameNet*. To overcome this difficulty, we are developing a formalized version of *FrameNet*, called *Re-FrameNet*, by rewriting part of *FrameNet* knowledge in a formal meta-language, which can be automatically translated into various planning languages.

In *Re-FrameNet*, a Frame in *FrameNet* is formalized as a 'meta-task', which is re-defined by a set of precondition, postcondition, invariant, and/or steps over semantic roles of the meta-task. In the definition, FEs (i.e., semantic roles) such as *Theme*, *Source*, and *Goal* of the Frame are taken as meta-variables. Therefore, the definition of a meta-task specifies the common semantic structure of all action verbs in the corresponding Frame. For example, the meta-task *put-placing* is defined like this:

(**define** (**meta-task** *put-placing*
(**:parameters** ?*Agent* ?*Theme* ?*Source* ?*Goal*))
(**:precondition** …)
(**:postcondition** …)
(**:invariant** …))

[1] https://framenet.icsi.berkeley.edu/fndrupal/

All of action verbs in Frame 'placing' (such as *lay*, *heap*, *deposit*) share the same definition. When a robot tries to understand an instruction with *put-placing* as its action verb (verb sense), our NLP components try to extract appropriate entities for every semantic roles specified in the definition of meta-task *put-placing*. This is the major mechanism of handling underspecification of naturalistic instructions in our prototype.

As a proof of concept, we built a first mini-version of *Re-FrameNet*[2], which contains 43 meta-tasks. We also developed an automated translator that converts these definitions into ASP language processible by our planner.

4.2 Task Planners

In the experiments we employed two planners based on HTN and ASP, respectively. HTN planning can use knowledge of task decomposition efficiently under some conditions [15]. Since OMICS provides such knowledge, we integrated this technique into our prototype. However, most of tasks from OMICS cannot be decomposed eventually into primitive actions of a robot and thus planned by an HTN planner, because many steps in OMICS are referred to by common verbs and OMICS does not contain decomposition knowledge for them. For example, *take*, *place*, *put*, *get*, and *turn* frequently occur in steps, while there is no tuple to decompose these tasks. It is the case of other on-line resources, since these action verbs refer to manipulations that are difficult to be described or specified by non-experts.

Fortunately, *FrameNet* (and other semantic dictionaries) provides rich knowledge about common verbs. We discovered that the *FrameNet* definition of an action verb can be reorganized by a set of precondition, postcondition, and invariant over semantic roles of the action (called the functional definition of action). But generally no decomposition knowledge of actions can be obtained from *FrameNet*. Therefore, we need some classical planner, which can utilize formalized functional definition of an action to plan the action. We implemented a classical planner in Answer Set Programming (ASP) [16], [17]. We note that this is the second discovery that motivated this effort and development of *Re-FrameNet*.

4.3 Skills and Action Models

A classical planner needs an action model as the interface between task planner and lower-level system, including motion planners. An action model consists of primitive actions. Each primitive action *a* is defined by a set of precondition, postcondition and

[2] http://ai.ustc.edu.cn/en/research/reframenet.php

invariant, just like the definition of a common verb in *Re-FrameNet*. They are conditions under which *a* can be executed, conditions that hold when *a* finishes, and conditions that must be satisfied during the execution of *a*. We assume in this paper that a primitive action is actually the formal specification of a robot skill, so that the planner can make use of primitive actions to generate a plan for a complex task, where a plan is a sequence of primitive actions. Of course, the more primitive actions a robot possesses, the more powerful it is. Given this, therefore, in this paper we do not consider the influence of primitive actions on robot performance, but focus on some other factors.

4.4 Experimental Results

We have conducted a series of experiments to evaluate the proposed approach on the prototype, where two large test sets collected from OMICS were used. We took a version of the OK-KeJia prototype that contains only two sorts of built-in knowledge, the action models and the semantically annotated lexemes of words in the lexicon, a type of linguistic knowledge which is only used in the multi-mode NLP module. The open knowledge the robot gathered for the task planning in the experiments was limited to two tables of OMICS and the synonymies of WordNet,[3] without any handcrafted knowledge.

Each test varied in either of the action model and the open knowledge base. Five action models, AM_1 = {*move*}, AM_2 = {*move*, *find*}, AM_3 = {*move*, *find*, *pick_up*}, AM_4 = {*move*, *find*, *pick_up*, *put_down*}, and AM_5 = {*move*, *find*, *pick_up*, *put_down*, *open*, *close*}, were chosen in order to examine the impact of the different action capabilities of a robot on its overall performance. Each test consisted of three rounds with different open knowledge bases, in order to show the impact of open knowledge on performance.

The task set in Test 1 contained 11,615 different tuples in *Tasks/Steps*, each consisting of a task name *T* and a sequence of steps (sub-tasks), *s*. In the first round of Test 1, only the action models were used in the planning procedure, with no open knowledge. Every task in the task set was input into the prototype to fulfill it. A task was solvable by the robot's planner if and only if each step of the task was a primitive action. In the second round, 11,615 *Tasks/Steps* tuples were used as a sort of procedural knowledge. In the third round, the robot tried to get open knowledge from both *Tasks/Steps* and WordNet. Consequently, some tasks such as "*move to a location*" and "*go to a location*" were identified as equivalent and executable by the prototype, although only the former could be matched to the robot's primitive action. Obviously, the open knowledge used in this test was extremely sparse—only the definitions of the tasks were used as the procedural knowledge.

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

Table 1: Experimental results

Open Knowledge	**AM$_1$**	**AM$_2$**	**AM$_3$**	**AM$_4$**	**AM$_5$**	**Percentage on AM$_5$**
Test 1 (11,615 user tasks)						
Null	6	24	45	164	207	1.78%
Tasks/Steps (11,615 rules)	7	28	51	174	219	1.89%
Tasks/Steps+WordNet	16	43	71	233	297	2.56%
Test 2 (467 user desires)						
Null	0	1	1	4	4	0.86%
Help+Tasks/Steps (15,020 rules)	29	63	83	107	117	25.05%
Help+Tasks/Steps+WordNet	43	73	87	119	134	28.69%

The task set in Test 2 consisted of 467 different desires that appeared in the *Help* table of OMICS, with duplicate ones discarded. Some examples of *Help* tuples are <*are sick, by giving medicine*>, <*are cold, by making hot tea*>, and <*feel thirsty, by offering drink*>. In each tuple, the first element is taken as a user desire, while the second element, a task, is taken as a means to meet the desire, not as the definition of the desire. In the first round of Test 2, no open knowledge was used. In the second round, all 3,405 unduplicated tuples in *Help* were taken as functional knowledge and all in *Tasks/Steps* as procedural knowledge. WordNet synonymies were used in the third round.

The experimental results are shown in Table 1 [6]. On every action model in each round, the number of tasks or desires that were fulfilled by the robot is listed in the table. In addition, the percentages of fulfilled tasks or desires with respect to the size of the task sets on AM$_5$ are listed in the last column. We observed that the overall performance increased remarkably due to the use of a moderate amount of open knowledge. The percentage of fulfilled tasks increased from 1.78% to 2.56% in Test 1 with very sparse open knowledge of two types, and from 0.86% to 28.69% in Test 2 with a moderate amount of open knowledge of three types, respectively. The difference in the performance improvements in the two tests further reveals the significant function of the amount or "density" of the open knowledge. We also observed that a robot’s basic ability (primitive actions) was still a key factor for the overall performance. As expected, the robot met more user requests with more primitive actions in all the cases. More detailed examination indicates that there were many user requests that were not met due to the powerlessness of the action models used in the experiments, though KeJia robots can do more actions that were not contained in the action models.

We note that the performance increased further in our experiments when *Re-FrameNet* was used with a set of techniques developed for this use [18], showing bigger spaces of improvement due to utility of more appropriate open knowledge.

5 Conclusion

As rapid development of intelligent robots, researchers are more and more interested in building robots that can work in large domains, such as those robots that are able to perform user tasks that have not been preprogrammed and are expressed naturally. Our analysis in this paper reveals that these requirements cannot be met using specified representations of robots or domains. Meanwhile, underspecified representations suggest new opportunities for development of such robots. In this paper, we propose the thesis of underspecification that underspecified representations should be employed to model robots working in large domains or interacting naturally with users. The true point here is not that incomplete representations of knowledge about a large domain or a large set of naturally expressed user tasks must be avoided, but that such knowledge should be incompletely represented, i.e., modeled in underspecified representations, in order to maintain consistency, competence and scalability of the knowledge. The thesis sets up a basis for our effort to build such robots. The feasibility of this open knowledge approach is preliminarily demonstrated by considering the knowledge rehabilitation problem. We describe an implemented robot prototype, OK-KeJia, and experiments on large test sets with it. The experimental results are encouraging.

Acknowledgments

The author thanks Jianmin Ji, Jiongkun Xie, Dongcai Lu and all other group members working for the KeJia Project, which has been supported by the National Hi-Tech Project of China under grant 2008AA01Z150 and the Natural Science Foundation of China under grants 60745002 and 61175057, as well as the USTC Key Direction Project and the USTC 985 Project. The author also thanks Michael Gelfond, Manuela Veloso, Xihong Wu, Beihai Zhou and Shlomo Zilberstein for their comments on this effort and earlier versions of this paper.

References

1. R. Cantrell, K. Talamadupula, P. Schermerhorn, J. Benton, S. Kambhampati, & 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* (HRI-12), Boston, USA, 2012, pages 471–478.
2. X. Chen, J. Ji, J. Jiang, G. Jin, F. Wang, J. Xie. Developing high-level cognitive functions for service robots. In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems* (AAMAS-10), Toronto, Canada, pages 989–996.
3. 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* (IROS-09), St. Louis, MO, USA, 2009, pages 4261–426.
4. W. Burgard, A. Cremers, D. Fox, D. Hahnel, 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.
5.T. Fong, C. Thorpe, and C. Baur. Robot, asker of questions. *Robotics and Autonomous Systems*, 42(3-4):235–243, 2003.
6. Xiaoping Chen, Jiongkun Xie, Jianmin Ji, and Zhiqiang Sui, Toward Open Knowledge Enabling for Human-Robot Interaction, *Journal of Human-Robot Interaction*, Vol. 1, No. 2, 2012, pages 100–117.
7. E. Davis, Naive physics perplex, *AI Magazine* 19(4), 1998.
8. Xiaoping Chen, Jianmin Ji, Zhiqiang Sui, Jiongkun Xie. Handling Open Knowledge for Service Robots. In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence* (IJCAI-2013), pages 2459-2465. 2013.
9. R. Gupta and M. Kochenderfer, Common sense data acquisition for indoor mobile robots, In: *Proceedings of the National Conference on Artificial Intelligence*, pages 605-610, Menlo Park, CA; Cambridge, MA; London; AAAI Press, MIT Press, 2004.
10. T. Kollar, S. Tellex, D. Roy, N. Roy, Toward understanding natural language directions, *5th ACM/IEEE International Conference on Human-Robot Interaction* (HRI), 2010.
11. S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, N. Roy, Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence* (AAAI-2011), 2011.
12. Jianmin Ji and Xiaoping Chen. A Weighted Causal Theory for Acquiring and Utilizing Open Knowledge. *International Journal of Approximate Reasoning* 55(9), 2014, 2071–2082.
13. J. Xie, X. Chen, Z. Sui, Translating Action Knowledge into High-Level Semantic Representations for Cognitive Robots, *Tenth International Workshop on Nonmonotonic Reasoning, Action and Change (NRAC 2013), Beijing, China*, Aug 5, 2013.

14. C. F. Baker, C. J. Fillmore, and J. B. Lowe. The berkeley framenet project. In: *Proceedings of the 17th international conference on Computational linguistics*-Volume 1, Association for Computational Linguistics, 1998, pages 86-90.
15. K. Erol, J. A. Hendler, and D. S. Nau. HTN planning: Complexity and expressivity. In: *Proceedings of the 12th National Conference on Artificial Intelligence* (AAAI-94), 1994, pages 1123-1128.
16. Gelfond, M., and Lifschitz, V. The stable model semantics for logic programming. In: *Proceedings of the 5th International Conference on Logic Programming* (ICLP-88), 1988, pages 1070–1080.
17. M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. clasp: A conict-driven answer set solver. In: *Proceedings of the 9th International Conference on Logic Programming and Nonmonotonic Reasoning* (LPNMR-07), 2007, pages 260-265.
18. D. Lu and X. Chen. An Integrated Prototype for Human-Robot Interaction: Techniques and Experiments, Technical Report, AI Research Center, USTC, 2014.