# Research on the Implementation of Chinese Instruction Parsing for Household Service Robots: Postprint

**Authors:** Xu Yuanyuan, Kong Lingfu, Gao Shengnan

**Date:** 2018-05-20T00:00:00+00:00

## Abstract

To enable household service robots to autonomously execute service tasks implied in Chinese instructions without human intervention, we propose a Chinese instruction task planning method based on answer set programming, which applies chunk annotation and answer set programming (ASP) to household service robot task planning. First, Chinese instructions are preprocessed through chunk annotation, then key information is converted into predicate sets according to transformation rules and rewritten as ASP rules. Furthermore, experimental results for each stage of Chinese service instruction processing are presented, and the mapping process from predicate sets to executable action sequences for robots is demonstrated with examples. Finally, the answer set is improved by merging partial atomic actions, which enhances solving efficiency, and cost planning is incorporated into task planning to ensure obtaining optimal action sequences. This method holds significant importance for promoting the development of natural human-robot interaction technology.

## Full Text

## Preamble

**URL**: http://www.arocmag.com/article/02-2019-01-029.html
**Journal**: ChinaXiv Partner Journal *Computer Application Research*
**Title**: Study on Chinese Instruction Parsing Issues for Home Service Robots
**Authors**: Xu Yuanyuan, Kong Lingfu, Gao Shengnan
**Affiliation**: Key Laboratory for Computer Virtual Technology & System Integration of Hebei, School of Information Science & Engineering, Yanshan University, Qinhuangdao, Hebei 066004, China

**Abstract**: To enable home service robots to autonomously execute service tasks implied in Chinese instructions without human intervention, this paper proposes an Answer Set Programming (ASP)-based task planning method for Chinese instructions, applying chunk tagging and ASP to home service robot task planning. First, Chinese instructions are preprocessed through chunk tagging. Then, key information is converted into predicate sets according to transformation rules and rewritten as ASP rules. Furthermore, experimental results for each stage of Chinese service instruction processing are presented, and examples illustrate the mapping process from predicate sets to executable robot action sequences. Finally, the answer set is improved by merging partial atomic actions, enhancing solving efficiency. Cost planning is incorporated into task planning to ensure optimal action sequences are found. This method holds significant importance for advancing natural human-robot interaction technology.

**Keywords**: answer set programming; task planning; chunk tagging; predicate sets

## 0 Introduction

With the rapid development of artificial intelligence technology in the coming years, an increasing number of robots will enter ordinary households to care for the elderly, children, and people with disabilities, thereby improving their quality of life. To better achieve natural human-robot interaction (HRI), leveraging human-human interaction patterns through natural language task instructions represents an ideal and effective HRI approach. Consequently, Chinese instruction task planning for home service robots has become an important research topic in the HRI domain.

Currently, research on HRI languages in the home service robot field predominantly focuses on English, with considerable progress made in English instruction task planning. In contrast, research on Chinese instruction task planning lags behind. For native Chinese speakers, not everyone can communicate with home service robots in English. Therefore, this paper conducts a series of studies on Chinese instruction task planning problems. In this field, the University of Science and Technology of China has achieved significant progress using Answer Set Programming (ASP) for English instruction task planning. In references [12–15], Chen Xiaoping et al. proposed a home service robot architecture that integrates ASP with Natural Language Processing (NLP) to achieve communication between humans and autonomous agents. Compared with current HRI methods primarily based on speech recognition and semantic identification technology, this approach offers better extensibility and powerful planning and reasoning capabilities. Reference [14] presents methods to improve ASP solving efficiency from a program conclusion perspective, enhancing the automatic task planning capability of home service robots. The "KeJia" robot demonstrated excellent performance in simulation tests, autonomously completing basic and complex tasks while acquiring causal knowledge from human-robot dialogue, proving ASP can effectively implement partial cognitive functions for service robots. In reference [17], Gao Shengnan et al. designed a service task analysis template that maps Chinese service instructions to action sequences, verifying the method' s effectiveness and feasibility through simulation experiments.

In summary, using ASP to solve English instruction task planning problems can yield action sequences, but low solving efficiency and how to obtain optimal planning remain existing issues. Moreover, compared with English instructions, Chinese instruction parsing presents greater difficulty in both preprocessing and task planning for three main reasons: (a) domestic and international scholars began researching English instructions earlier with mature methods, while Chinese instruction research started later, and English instruction processing methods are not entirely applicable to Chinese; (b) Chinese instruction expression is relatively casual, with one semantic meaning potentially having multiple expression forms; (c) the same expression in Chinese instructions may have different semantics in different contexts. The service task parsing template method used in reference [17] handles relatively fixed types of Chinese service instructions with insufficient flexibility. To address these problems, this paper improves ASP to make it suitable for Chinese service instruction task planning processing. By merging some atomic actions, the ASP instantiation solving process is simplified, improving ASP solving efficiency. Additionally, this paper attempts to clarify the minimum cost of executing action planning, i.e., finding optimal planning, to better realize the Chinese service instruction task planning process.

## 1 Home Service Robot System Architecture

To address the research problem of Chinese instruction task planning for home service robots, this paper proposes an autonomous processing architecture for Chinese instructions in home service robots. As shown in [Figure 1: see original paper], the Chinese service instruction parsing process is divided into two modules.

The instruction preprocessing module comprises chunk tagging, predicate set conversion, and a language knowledge base (Language KB). This layer's primary function is to understand user-robot dialogue in Chinese instructions, preparing for task planning through NLP techniques such as chunk tagging and predicate conversion.

The task planning module includes task information, scene information, ASP rules, sensors, and a domain knowledge base (Domain KB). The robot can obtain domain knowledge required for instruction processing from the Domain KB, such as distances between locations and relationships between objects. Users parse task instructions to obtain scene and task information, write ASP rules, call the iclingo solver, and finally obtain a set of executable action sequences for the robot.

## 2 Chinese Service Instruction Preprocessing

### 2.1 Chinese Service Instruction Corpus

This research group has collected a small-scale corpus to provide an experimental platform for related research. Overall, the collected Chinese instructions are frequently issued service commands by users in home environments, with relatively simple structures but containing key information required for robot execution, such as objects to be manipulated, their placement locations, and service recipients. Instructions A, B, and C contain different forms of task information, while D-class instructions are special, containing not only task information but also environmental description information. Among them, Class A instructions are robot-human-object interaction instructions; Class B are robot-object-location interaction instructions; Class C are robot-object interaction instructions. Four typical classes of Chinese service instructions are as follows:

**Class A**: (1) Give me a bottle of mineral water; (2) Give Anne an apple
**Class B**: (1) Put the bowl on the dining table; (2) Move a chair in front of the TV
**Class C**: (1) Close the living room door; (2) Turn on the TV
**Class D**: (1) Give Jack an apple, the apple is on the table; (2) Put the book on the desk, give Tom a glass of water

### 2.2 Syntactic Analysis

Drawing on reference [19], this paper employs chunk tagging for syntactic analysis of Chinese instructions. Due to space limitations, this paper only performs preprocessing operations on the first service instruction of each class. The basic chunks in service instructions include target chunk (tc), service chunk (sc), location chunk (lc), and action chunk (ac). If basic chunk rules are not satisfied, they are labeled as null.

The syntactic analysis process for the four typical Chinese instructions is performed with respect to four object types, as shown in –.

** Syntactic Analysis Process for Class A Instruction**
*Give me a bottle of mineral water*
Part-of-speech tagging: 给/v, 我/r, 一/m, 瓶/q, 矿泉水/n
Chunk tagging: 给/ac, 我/sc, 一瓶矿泉水/tc

** Syntactic Analysis Process for Class B Instruction**
*Put the bowl on the dining table*
Part-of-speech tagging: 把/p, 碗/n, 放/v, 在/p, 餐桌/n, 上/f
Chunk tagging: 把/null, 碗/tc, 放/ac, 在/null, 餐桌上/lc

** Syntactic Analysis Process for Class C Instruction**
*Close the living room door*
Part-of-speech tagging: 关/v, 客厅/n, 的/u, 门/n
Chunk tagging: 关/ac, 客厅的门/tc

** Syntactic Analysis Process for Class D Instruction**
*Give Jack an apple, the apple is on the table*
Part-of-speech tagging: 给/v, Jack/nr, 一/m, 个/q, 苹果/n, 苹果/n, 在/p, 桌子/n, 上/f
Chunk tagging: 给/ac, Jack/sc, 一个苹果/tc, 苹果/tc, 在/null, 桌子上/lc

### 2.3 Predicate Transformation Rules

For robot-oriented Chinese instruction task planning problems, the first step is understanding task information in Chinese instructions. The Chinese instructions processed in this paper are mostly simple natural language with restricted vocabulary, making preprocessing relatively straightforward. The previous section implemented part-of-speech tagging and chunk tagging for Chinese instructions, obtaining keywords and semantic chunk relationships. The target chunk semantically corresponds to objects the robot must manipulate; the service chunk represents people requiring service; the location chunk indicates where objects should be placed; and the action chunk represents actions the robot needs to perform on target chunks.

This paper' s predicate transformation method is similar to reference [18] in solving how to convert key information into predicate sets. However, while reference [18] only converts keywords appearing in instructions into corresponding predicate sets, this paper's method maps the combination relationships between keywords and semantic chunks to predicate sets. For example, only the combination of the keyword "去/到" (go/to) and a location chunk can correspond to the predicate set move(X), where X represents the location information indicated by the location chunk. The combination of the keyword "放入/放到" (put into/put on) and its preceding and following two target chunks corresponds to the predicate set putin(objA, objB), where objA and objB represent the two target chunk information. This makes transformation rules more rigorous and accurate, avoiding errors caused by simple correspondence.

Through the combination relationships between keywords and various semantic chunks in the above Chinese instructions, rules for converting keywords and semantic chunks into predicate sets are obtained, as shown in .

** Predicate Set Transformation Rules**
Keywords and Semantic Chunks | Predicate Set
–|–
去/ac+lc | move(X)
给/ac+sc | give(human, objA)
拿起/ac+tc | pickup(objA)
tc1+ 放在/ac+tc2 | putdown(objA, objB)
tc1+ 在/null+tc2+ 上/lc | on(objA, objB)
放下/ac+tc | putdown(objA)
关闭/ac+tc | close(objA)
打开/ac+tc | open(objA)

tc1+ 放入/ac+tc2 | putin(objA, objB)
tc1+ 取出/ac+tc2 | takeout(objA, objB)
抓住/ac+tc | catch(objA)

## 3 Chinese Service Task Planning

### 3.1 Chinese Service Instruction Flowchart

In a home environment, Chinese instructions for robots consist of scenario description and task description. Scenario description determines the initial state of the home environment, providing information including object types, locations, and other attributes in the scene, as well as the robot' s current state. Task description contains detailed information about user-assigned tasks, including one or more tasks, constraints, and other information. For example, in Class D service instruction "Give Jack an apple, the apple is on the table," the former is a description of the task assigned by the user, while the latter describes the home environment scenario. The flowchart for solving Chinese service instruction task sequences using ASP is shown in [Figure 2: see original paper].

As shown in [Figure 2: see original paper], the process of solving action sequences for Chinese instructions based on ASP is as follows: (a) Determine information in the instruction. If it is task information, process the task information and add it to the ASP program in the form of facts or rules; if it is scene information, process the scene information and convert it to predicate sets according to predicate transformation rules. (b) Based on ASP solving status and initial/target states, solve the ASP program by calling the iclingo solver. (c) Obtain the optimal action sequence. If time exceeds 5 seconds, a timeout is displayed, requiring re-planning of scene information to check for incorrect or missing information until an optimal sequence is obtained.

### 3.2 Chinese Service Task Planning

Task planning aims to map Chinese instructions to optimal executable action sequences for robots based on environmental description, initial/target states, and atomic action-related content. First, service instructions are rewritten into state descriptions after instruction completion, and these state descriptions are added to the ASP program as constraints. During robot operation, the Chinese instruction preprocessing module converts user instructions into internal logical formulas, and the task planning module adds these formulas to the ASP program as facts, activating corresponding constraints, ultimately forming a satisfied target state. On this basis, computing answer sets is equivalent to obtaining robot action sequences from initial to target states through classical task planning methods. Other information is also added to the ASP program as facts or rules in the above manner, using domain knowledge to derive more knowledge from this information [12].

ASP serves as a general tool for knowledge representation and reasoning with non-monotonic reasoning capabilities [12]. It can characterize robot internal information, including robot perception of the environment and its action capabilities, convert external input information such as natural language into corresponding ASP rules, and use this information for parsing and diagnosis to solve Chinese service instruction task planning problems. ASP solving includes two steps: (a) instantiation, which instantiates variables in the program into corresponding instances; (b) model search, which computes answer sets of the instantiated program. This paper uses iclingo, an incremental ASP solver based on clasp and Gringo methods that incrementally solves logic programs using a partitioning sequence approach. Since in practical problems, separately computing answer sets for the preceding part may be too numerous, making the partitioning sequence method inefficient, iclingo no longer directly computes answer sets for the preceding program but instantiates the subsequent program based on estimated atoms.

Each subtask in task planning is called an ASP action. Actions and environmental features are represented by predicate sets in ASP. For example, in a home environment, on(A, B, T) indicates that small object A is on object B at time T; location(A, X, T) indicates that small object A is at location X at time T; holding(A, T) indicates that object A is in the robot' s gripper or plate at time T. For instance, the robot' s "move" action and the predicate "location" are expressed as follows:

### 3.3 Cost Planning

In the previous section, service instructions were rewritten as ASP rules, and calling the solver could yield planning of varying sizes, with any one selected for execution. However, in practice, these plans are not equivalent because different actions have different costs in real life (such as time spent, energy consumed, work done, etc.). This paper takes the time used in action execution as the primary factor for cost planning.

This section demonstrates how to combine cost with actions to select a plan with minimum cost to achieve the goal. Cost is defined as a function containing two parameters: the action being completed and the state when the action begins. For simplicity in this project, all actions except move are assumed to have the following fixed costs:

Now defining the cost of the move action, which depends on the robot' s initial position before executing the action. The cost of moving from location Y to location X is calculated as follows:

.),( )(),(),1),,((),),,(()1),,((.),(),(),( ),),,((),),((.),(),(),( ),),,(()1),,((lasttimeTT-timeXnumberAobjTXAlocationholdsnotTXAlocationholdsTXAlocationholdslasttimeTTtimeXnumberZagentT
.),( ),(),(),(),,( : ),2(.),( ),(),(),(),,( : ),2(.),(),(),(close : ),1(.),(),(),(open : ),1(.),(),(),( : ),1(.),(),(),( : ),1(.),(),(),( : ),2(.),(),(),( : ),2(. ),(),(),,(

: ),1(lasttimeTBcontainerBobjAobjAsmallBAputupTcostlasttimeTBcon-tainerBobjAobjAsmallBAputinTcostlasttimeTBcontainerBobjBTcostlasttimeTBcontainerBobjBTcostlasttime

Based on the above cost planning, it can be found that optimal planning is not necessarily the shortest planning. In cost planning, iclingo only needs to be called once, with the increment n directly assigned as max_len. Using the above optimization statement, this paper finds the optimal answer set according to iclingo, with the optimal planning size within max_len.

### 3.4 Human-Robot Interaction Task Planning

The robot has a perception action. If the robot does not know the location of a person X or an object A, the robot can ask the user for their location information at time T. The action askloc(A, T) indicates that the robot asks for the location of object A at time T. The action askloc(X, T) indicates that the robot asks for the location of person X at time T. When performing task planning, the robot can obtain object locations through human-robot dialogue. After execution, location answers about a person or object are obtained, with ASP rules as follows:

## 4 Examples

### 4.1 Environmental Information

This section uses four typical Chinese service instructions as examples to solve for optimal action sequences using ASP. Environmental information is stored in the knowledge base in the following form: objects in the home environment and robot locations are assigned positive integers. For simplicity, the number 0 represents "nothing," and the number 1 indicates the robot is at location 1. Different locations are represented by non-negative integers. Robot and object information in the home environment is as follows:

#### 4.2.1 Solving Class A Instructions with Answer Sets

In Class A instructions, "Give me a bottle of mineral water," through preprocessing operations, keywords and semantic chunks can be obtained. According to predicate set transformation rules, it can be converted into the following predicate set form:

Added to ASP rules as facts, along with two corresponding constraints:

Combined with scene description information from the domain knowledge base, the following action sequence is solved:

#### 4.2.2 Solving Class B Instructions with Answer Sets

In Class B instructions, "Put the bowl on the dining table,"converted to predicate set form:

Added to ASP rules as facts, along with two corresponding constraints:

Under the same conditions, the action sequence is obtained as:

#### 4.2.3 Solving Class C Instructions with Answer Sets

In Class C instructions, "Close the living room door," converted to predicate set form:

Added to ASP rules as facts, along with its corresponding constraints:

Similarly, the action sequence is solved as:

#### 4.2.4 Solving Class D Instructions with Answer Sets

In Class D instructions, "Give Jack an apple, the apple is on the table," contains not only one piece of task information but also one piece of scene description information. According to predicate set transformation rules, it is converted into the following two predicate set forms:

The task planning module converts the task description "Give Jack an apple" into the predicate , added to the ASP program as a fact, along with two corresponding constraints:

The scene description "The apple is on the table" is converted to predicate , added to the ASP program as a fact. The domain knowledge base typically also contains other rules expressing domain knowledge, for example:

The latter two rules indicate that if small object A is on object B, then object A' s location is the same as object B' s location. From the service instruction "The apple is on the table," it is concluded that the apple and table are at the same location.

From the domain knowledge base, the following knowledge can be obtained: object 18 is an apple, the apple is on the table and the table is at location 10, Jack' s location is 3, and the robot' s location is 1. By default, any two locations are reachable. When the quantity of target objects is not explicitly stated, it defaults to 1. By computing the answer set of the ASP program, the action sequence can be obtained:

Where 18 represents the ID of "apple." The entire process is: first move to location 10, then catch object 18 (the "apple" ), then move to location 3, and finally give the held object to Jack. This ASP program' s answer set contains one action sequence, which represents the robot' s action steps to complete this instruction.

### 4.3 Macro-Action Answer Set Solving

After years of research by domestic and international scholars, answer set theory has matured. However, due to the instantiation enumeration process, ASP solving efficiency remains low, limiting its widespread application. In the previous section, four typical service instructions were solved using ASP to obtain instruction execution sequences. Observation reveals that task sequences contain a large number of move actions, affecting task planning efficiency. Accordingly, this project merges certain regular action instructions at the atomic level before planning, effectively reducing task planning time and improving solving efficiency. For example, in Class A instructions, before picking up mineral water, the robot must be at the same location as the mineral water, and when giving the mineral water to "me," the robot must be at the same location as "me." Therefore, before task planning, the move(A) action and catch(A) action can be merged into a macro-action mcatch(A). The corresponding ASP rule is:

Similarly, the move(X) action and give(human, A) action can be merged into the mgive(A) action, with corresponding ASP rules:

When the robot manipulates a container object, such as needing to put object A into container B, the execution conditions require the robot to be at the same location as container B. Before retrieving the object, the container door must be opened by executing open(B), then object A is put into container B by executing putin(A, B), and finally the container door is closed by executing close(B). Therefore, the above four actions can be merged into two composite actions mopen(B) and piclo(A, B), with corresponding answer set rules:

Through the above merging operations of partial atoms in ASP, the instantiation process can be reduced, improving ASP solving efficiency and thereby reducing the time for service instruction task planning.

# 5 Experimental Results and Analysis

## 5.1 Home Service Robot Simulation Platform

The simulation platform simulates home service robots as 3D simulation robots using a home environment as the test environment. Information obtained by the robot through sensors and through human-robot dialogue is represented as scene information. User task instructions issued to the robot in Chinese are described as task information. The platform adopts a client-server model, where the server provides environment description and scoring functions. The client program performs task planning based on this information within a specified time (not exceeding 5 seconds), returns the planning results to the server, and evaluates the obtained action sequence.

## 5.2 Chinese Service Instruction Task Planning Experiment

Based on different Chinese instruction types, this paper selects four typical service instructions for experimental verification on the home service robot simulation platform. To test the improved ASP solving efficiency, this paper' s method is compared with the method in reference [6]. The obtained execution time and cost planning can intuitively demonstrate the effectiveness of this paper' s method.

In Class A instruction "Give me a bottle of mineral water," the execution condition is that the robot has a bottle of mineral water in its gripper and is at the same location as "me." Therefore, before executing give(me, water), the robot must first move to the mineral water' s location, then catch a bottle of mineral water, then move to "my" location, and finally give the mineral water to "me." From this description, when manipulating target objects in service instructions, the robot must first move to the target object' s location. During task planning, a large number of move actions in the planning sequence severely affect ASP efficiency. Therefore, this paper merges the move(A) action with the next service action, reducing task planning time and improving ASP solving efficiency. The other three instruction types undergo similar partial atomic action merging operations.

This paper selects four representative service instructions from the four classes and solves them using both the simple ASP method from reference [6] and the improved ASP method. The results are shown in .

** Experimental Comparison Results**

| Instruction | ASP Program Answer Set | Improved ASP Program Answer Set |
|---|---|---|
| Give me a bottle of mineral water | occurs(move(10), 1), occurs(catch(5), 2), occurs(move(2), 3), occurs(give(2,5), 4) | occurs(mcatch(5),1), occurs(mgive(2,5),2) |
| Put the bowl on the dining table | occurs(move(6), 1), occurs(pickup(13), 2), occurs(move(10), 3), occurs(putdown(13), 4) | occurs(mpickup(13),1), occurs(mputdown(13),2) |
| Close the living room door | occurs(move(13),1), occurs(close(23),2) | occurs(mclo(23),1) |
| Give Jack an apple, the apple is on the table | occurs(move(10),1), occurs(catch(18),2), occurs(move(3),3), occurs(give(3,18),4) | occurs(mcatch(18),1), occurs(mgive(3,18),2) |

As shown in , both methods can obtain an action sequence, and both sequences are executable by the robot. However, [Figure 3: see original paper] shows that the improved ASP method significantly reduces task planning time. This paper studies planning for single Chinese instructions each time, where experimental results do not show obvious differences, but when planning multiple instructions simultaneously, reducing a large number of move actions would greatly improve execution efficiency.

Ten experiments were conducted on Class A service instructions using both methods on the home service robot simulation platform. Service instruction planning times are shown in [Figure 3: see original paper].

**[Figure 3: see original paper] Comparison of Solving Time Between Simple ASP Method and Improved ASP Method**

As shown in [Figure 3: see original paper], the improved ASP method' s solving time for Class A instructions is approximately 0.3–0.4 seconds, while the unprocessed ASP method’ s solving time varies significantly, approximately 0.7-0.9 seconds. In the 3rd and 8th experiments, task planning time was longer, mainly due to missing or incorrect information obtained from the domain knowledge base. This demonstrates that improving ASP by merging partial atoms can effectively enhance ASP solving efficiency.

Experimental results prove that the ASP method can plan executable action sequences for home service robots and that the proposed method of merging partial atomic operations helps improve ASP efficiency in Chinese instruction task planning.

### 5.3 Chinese Service Instruction Cost Planning Experiment

This section incorporates cost planning into ASP. By calculating the cost of action sequences, the planning result with minimum cost can be obtained, making the resulting task planning sequence optimal. Additionally, a perception action askloc(A, X) is added to obtain object or person locations through human-robot interaction.

In the home simulation environment, the robot and service recipient Tom are in the same room, while objects A and B are in another adjacent room. The scene information is as follows: robot(1), location(1, 1), Tom(2), location(2, 2), A(3), location(3, 3), B(4), location(4, 4).

For task instructions “Give object A to Tom” and “Give object B to Tom,” object and environment location information is shown in [Figure 4: see original paper] and [Figure 5: see original paper].

Based on the above instruction task information and environment, two action plans can be obtained:

**Action Sequence (1)** has execution effects shown in [Figure 4: see original paper].
**Action Sequence (2)** has execution effects shown in [Figure 5: see original paper].

According to action cost specifications in Section 4.3, the cost planning for action sequences is calculated. For simplicity, the cost of move actions is specified as fixed cost in this experiment. The cost of action sequence (1) obtained from the above service instructions is 24. The cost of action sequence (2) is 22. Therefore, according to cost consumption, the action sequence with lower cost (2) is selected. This shows that optimal planning is not necessarily the shortest planning, and incorporating cost planning helps find the optimal planning sequence for service instructions.

## 6 Conclusion

Chinese instructions, due to their flexible grammar and casual structure, present certain difficulties for task planning. Therefore, this paper first performs preprocessing operations to simplify their structure and convert them into predicate sets according to predicate set transformation rules. The ASP method is improved to make it applicable to Chinese service instruction task planning research. Observing the final solving results and action sequences reveals that the obtained task sequences contain a large number of move actions, affecting solving efficiency. Consequently, this paper merges move actions with other execution actions in ASP through atomic operation merging, significantly improving solving efficiency. Additionally, some conventional instruction actions can also undergo atomic operation merging to shorten task planning time and enhance ASP solving efficiency. On this basis, cost planning is incorporated to calculate cost consumption of obtained action sequences, selecting action sequences with lower cost rather than shorter task planning as optimal action sequences. Finally, simulation experiments and results demonstrate the effectiveness of the proposed method.

## References

[1] Ren Fuji, Sun Xiao. Status and development of intelligent robots [J]. Science & Technology Review, 2015, 33 (21): 65-70.

[2] Kollar T, Perera V, Nardi D, et al. Learning environmental knowledge from task-based human-robot dialog [C]// Proc of IEEE International Conference on Robotics and Automation. 2013: 4304-4309.

[3] Duvallet F, Kollar T, Stentz A. Imitation learning for natural language direction following through unknown environment [C]// Proc of IEEE International Conference on Robotics and Automation. 2013: 1047-1053.

[4] Johnson D O, Agah A. Human robot interaction through semantic integration of multiple modalities, dialog management, and contexts [J]. International Journal of Social Robotics, 2009, 1 (4): 283-305.

[5] Wu Junhua, Liu Longxia. Chunk parsing and entity relation extracting to chinese text by using conditional random fields model [J]. Journal of Intelligent Learning Systems & Applications, 2010, 2 (3): 139-146.

[6] Wang Fan, Fan Wenbin, Zhang Xueyan, et al. Preliminary exploration of improving answer set programming solving efficiency in home robot simulation [J]. Journal of Chifeng University: Natural Science Edition, 2014, 30 (3): 19-21.

[7] Perera V, Veloso M. Handling complex commands as service robot task requests [C]// Proc of International Conference on Artificial Intelligence. [S. l. ] : AAAI Press, 2015: 1177-1183.

[8] Li Sujian, Liu Qun, Bai Shuo. Chinese chunk analysis combining statistics and rules [J]. Journal of Computer Research and Development, 2002, 39 (4): 385-391.

[9] Xie Jiongkun. Research on natural language understanding for human-robot interaction [D]. Hefei: University of Science and Technology of China, 2015.

[10] Jin Guoqiang, Chen Xiaoping. Action language extension for intelligent service robot task planning [J]. Journal of Software, 2013, 24 (7): 1614-1625.

[11] Ivanov K S. Action of robot with adaptive electric drives of modules [M]// Advances on Theory and Practice of Robots and Manipulators. [S. l. ] : Springer International Publishing, 2014: 563-569.

[12] Chen Xiaoping, Ji Jianmin, Jiang Jiehui, et al. A service robot architecture supporting personalized coordination [J]. Journal of Nanjing University: Natural Sciences, 2010, 46 (2): 131-139.

[13] Chen Xiaoping, Jiang Jiehui, Ji Jianmin, et al. Integrating NLP with reasoning about actions for autonomous agents communicating with humans [C]// Proc of IEEE//WIC//ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies. 2009: 137-140.

[14] Ji Jianmin. Several approaches to improving ASP efficiency and applications on service robots [D]. Hefei: University of Science and Technology of China, 2010.

[15] Chen Xiaoping, Xie Jiongkun, Ji jianmin, et al. Toward Open Knowledge Enabling for Human-Robot Interaction [J]. Journal of Human-Robot Interaction, 2013, 1 (2): 100-117.

[16] Cho J, Kim H, Sohn J. Implementing automated robot task planning and execution based on description logic KB [M]// Simulation, Modeling, and Programming for Autonomous Robots. Berlin: Springer, 2010: 217-228.

[17] Gao Shengnan, Kong Lingfu, Wu Peiliang. Autonomous processing method for Chinese service instructions for indoor intelligent robots [J]. Robot, 2015, 37 (4): 424-434.

[18] Zhang Boya. Research on Chinese voice instruction parsing for home service robots [D]. Qinhuangdao: Yanshan University, 2015.

[19] Li Xinde, Zhang Xiulong, Dai Xianzhong. A mobile robot visual navigation method based on restricted natural language processing [J]. Robot, 2012, 33 (6): 742-749.

[20] Yang F, Khandelwal P, Leonetti M, et al. Planning in answer set programming while learning action costs for mobile robots [C]// Proc of AAAI Symposium. 2014.

*Note: Figure translations are in progress. See original paper for figures.*