

Reasoning of Motion through Task Order for Teaching by Non-Professional User

Rie KATSUKI^{†‡}, Roland SIEGWART[†]

[†] STI-I2S-ASL

Swiss Federal Institute of Technology Lausanne (EPFL)
1015-Lausanne, Switzerland
rie@prince.pe.u-tokyo.ac.jp

Jun OTA[‡], Tamio ARAI[‡]

[‡] Dept. of Precision Engineering,
School of Engineering
The University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Abstract— This paper proposes a method that reasons intent of motion through task order such as “switch on the light.” Intent of motion means kinds of path of a contact point between a target and an end effector when a robot recreates ordered task. The intent is used for teaching. After a robot fixes a framework of task using the intent, it asks questions to a user to specify concrete values in the framework, the user replies them. This interaction creates motions of the robot. The kinds of path are defined as linear, circular and Point to Point. Moreover, impossible task is also considered. Four sets of $P_p(w)$: Probability that a word represents a path, and $C_p(w)$: Certainty of $P_p(w)$ are added to each word in an electric thesaurus to enable the reasoning. After values are input in part of the sets by 8 production rules, they are propagated to the other sets through hierarchical relationship of words in the thesaurus. When the user enters two words, namely task and target, $\text{argmax}_p\{P_p(w)*C_p(w)\}$ becomes answer. Moreover, update and re-propagation of values in the sets using answers from the user make the reasoning more precise.

Keywords-Reasoning; Semantics; Learning; Teaching

I. INTRODUCTION

Teaching by non-professional user is most important to expand an ability of robot to domestic environments. Therefore, teaching by demonstration[1] is a powerful method. To recreate a taught task, robot has to acquire many kinds of data. Furthermore, it has to grasp the intent of motions during demonstration to get rid of noise in demonstrated data and create motions that correspond to the location of a target. Naturally, non-professional user needs to instruct the robot about two items: when and what should the robot attend to? What kind of motion is being taught? It is hard for the user to figure out them.

Therefore, we propose a novel teaching method in which a robot asks values of the data needed for task realization. The robot already owns a framework of task, i.e. a flow chart of motions. Iteration of questions and answers between the robot and a user with keyboard input or partial demonstration builds the robot’s motions interactively. A detail of the framework is fixed by intents of the motions (“Intent” means a brief path. More detail is described in Section II). The intent is reasoned through a task order, which is formed by natural language for the sake of non-professional users. Consequently, a method to

reason the intent through rich natural language becomes an important issue.

Expert systems[2][3], a typical reasoning system based on natural language bring excellent results specially in the situation that needs professional knowledge like medical diagnosis. However, problems arise when the system is expanded to more general topics: reasoning without knowledge about the question, maintenance of knowledge. In robotics, Ahlrichs et al. have picked up items needed to move to a destination using a hand-made semantic network[4]. Wermter et al. have matched words regarding motion primitives and robot’s actions[5]. The conversion from the word to semantic expression was done by hand. There is no study that allows any natural language because the number of words and their ambiguities are hard problems.

In this study, we propose a system that reasons intent of motion through task order. To generalize input words, we apply a ready-made knowledge base that covers general purpose. To be able to reason with small amount of rules that are appended to the knowledge base, we utilize hierarchical relationships between the rules and an entered task order in the knowledge base. Furthermore, to realize more precise reasoning, the system learns correct answers from a user. An overview of the reasoning is presented in Section II. The knowledge building method and algorithm of reasoning are described in Section III. In Section IV, the algorithm of the learning is introduced. Reasoning results before and after the learning are compared in Section V. Conclusions follow in Section VI.

II. OVERVIEW

For the framework of a task, we defined five steps to execute a task based on Kuniyoshi[1].

1. A robot approaches to a grasp point on a target.
2. It grasps the target.
3. It moves together with the target.
4. It releases the target.
5. It returns to home position.

We call a flowchart consists of the above steps “framework”, as illustrated in Figure 1. Each step includes

a function, and then, its arguments mean concrete values to create a motion. A robot asks questions to a user for acquisition the values.

Each function is identified by “intent of motion,” which stands for a brief path of the contact point between an end effector and a target(hereinafter the point is referred to as “CPT”). Note that the path does not identify the concrete position of CPT. We define three kinds of the paths: *linear*, *circular* and *PTP*(Point to Point, a path that regards only initial and end position), since there are few tasks with complicated path in domestic environments. In every task, to execute step 1. or 5. the robot moves with *PTP*. In 2. and 4., it moves only its end effector. Therefore, the functions in 1., 2., 4. and 5. are fixed. On the other hand, the all paths can apply to 3.

Therefore, the system showed in Figure 2. reasons the path of 3. from a task order. The figure shows input and output dialog boxes. We consider that task orders form “task+target” such as “switch on the light”. Hence, the input of the system is two words: Task and Target. In the output dialog, “opinion” shows a result of reason; “reason” shows a reason of the result. The lower half of the dialog box is an interface for collecting correct paths from a user. The system updates its knowledge using the paths. Here, although user imagines *the* Task and *the* Target, the system reasons a path about *a* Task and *a* Target due to ambiguity of words(e.g. “door” with rotating or with sliding). *A* Task/Target is equal to a group of *the* Task/Target; it sometimes outputs more than two kinds of paths. In addition, the system addresses only tasks with a kind of path. When it addresses the task that is combined some kinds of paths, the tasks are regarded as chains of the tasks with a kind of path.

III. ALGORITHM FOR REASONING PATH

A. Building of Knowledge Base

The ideal knowledge base is a huge multi-purpose one

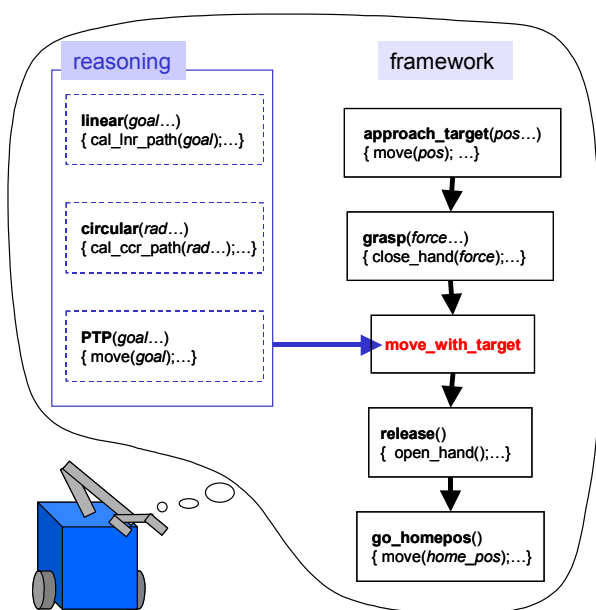


Figure 1. Framework of task

like Semantic Web[6]. However, The use of Semantic Web is too early because it is still in an initial stage that establishes its format[7]. Therefore, we substitute an electric thesaurus, which is WordNet[8]. Its design is inspired by current psycholinguistic and computational theories of human lexical memory. English nouns, verbs, adjectives, and adverbs are organized into synonym sets, each representing one underlying lexicalized concept. Different relations link the synonym sets.

WordNet has Hypernyms and Hyponyms. “Hypernyms of A” mean A is a kind of...(noun)/A is one way to...(verb), “Hyponyms of A” mean ...is a kind of A(noun)/...is a particular way to A(verb). When Hypernyms and Hyponyms are connected, they form trees. The structure of the trees corresponds to the relationships between concepts of the words. Therefore, we use the trees as knowledge base. The trees that consist of only verbs or nouns are applied since every Task or Target is verb or noun. The number of their nodes are 13508(verbs) and 79689(nouns).

B. Addition of Probabilities and Certainties

To enable reasoning with the trees, we define a set of $P_p(w)$:probability of path and $C_p(w)$:certainty factor for $P_p(w)$ for each word as shown in Figure 3., where suffix p is path, w is word. In a case of w =verb, i.e. w =Task, $P_p(w)$ stands for the frequency that CPT passes on p . For instance, $P_{ccr}(rotate)=1.0$ means that CPT passes on *circular* path every time(100%) when a robot executes *rotate*. On the other hand, In a case of w =noun, i.e. w =Target, $P_p(w)$ stands for the ratio of objects that intend the typical path p among objects w . Some objects can require different kinds of path, for example, doors can be rotating or sliding. If 70% of all doors are rotating doors, then $P_{ccr}(door)=0.7$. Moreover, $C_{ccr}(door)=0.5$ means that the system has moderate(50%) certainty for $P_{ccr}(door)=0.7$. $C_p(w)$ is applied to express approximate results and allows reasoning with uncertain knowledge[9].

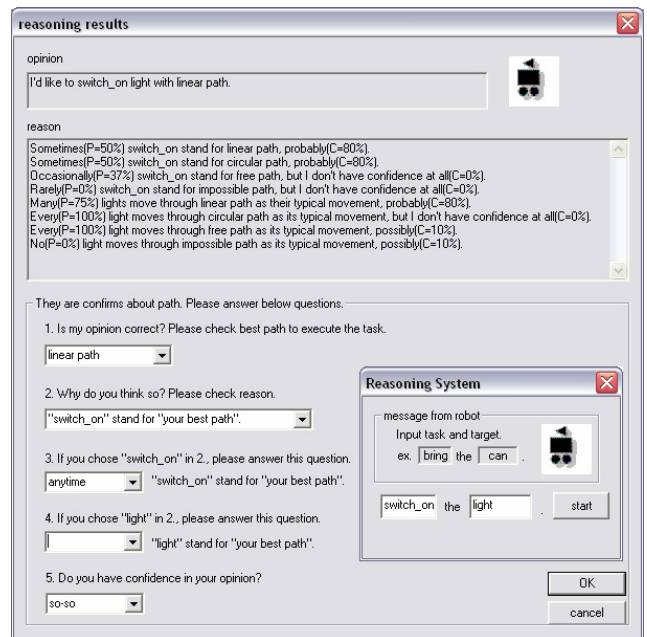


Figure 2. Input and output dialog boxes

order: switch on the light

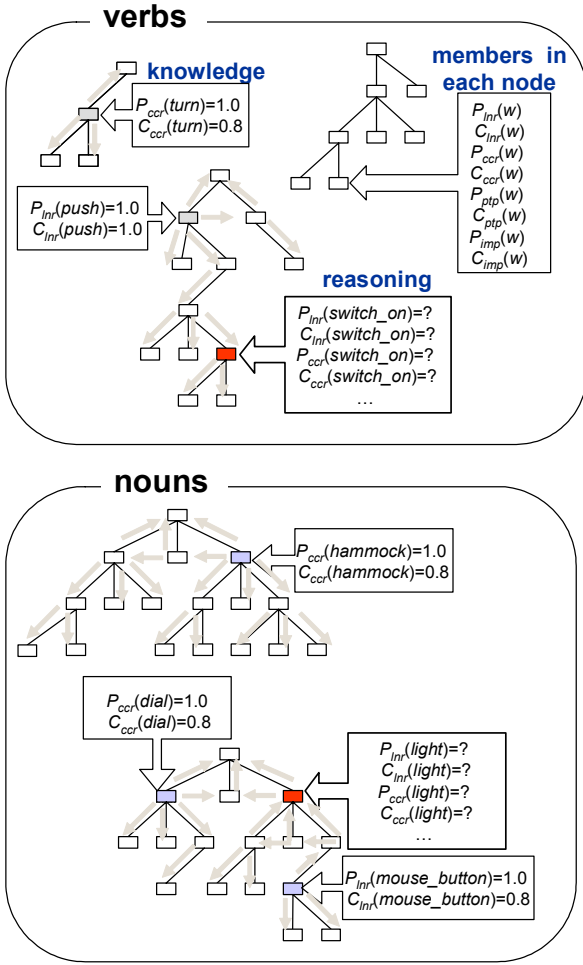


Figure 3. Concept of reasoning

For the sake of *linear*, *circular*, *PTP*, and *impossible*, four sets of $P_p(w)$ and $C_p(w)$ are set in each word: $P_{lnr}(w)$ and $C_{lnr}(w)$, $P_{ccr}(w)$ and $C_{ccr}(w)$, $P_{ptp}(w)$ and $C_{ptp}(w)$, $P_{imp}(w)$ and $C_{imp}(w)$. Using probabilities that are independent about the paths make the system possible to express ambiguities of words. For instance, “*open* stands for *linear* or *circular* depending on Target” can be expressed using both $P_{lnr}(open)$ and $P_{ccr}(open)$. *Impossible* means the robot cannot execute the task.

C. Setting values of Probabilities and Certainties

This subsection describes how values of $P_p(w)$ and $C_p(w)$ are set. First, we define production rules to assign initial values in $P_p(w)$ and $C_p(w)$. TABLE I. shows the rules and TABLE II. shows parts of the initial values. The number of the initial values is about 1000. Many words in TABLE I. have multiple meaning and they are distinguished by ID, so we judge each meaning whether it matches the rule or not using its glossary. Nevertheless, rules No.4-7 that pick up associated words through their glossary by a hand-made program cannot discriminate the ambiguity, therefore we set $C_p(w)=0.8$. Moreover, in rule No.8, Holonyms(i.e. A is a part of...) are used to assign $P_p(w)$ and $C_p(w)$.

TABLE I. PRODUCTION RULES

No.	Production Rules
1	IF Task is <i>slide</i> or <i>pull</i> or <i>push</i> THEN path is <i>linear</i> and NOT <i>circular</i> and <i>PTP</i> , and robot can execute the Task.
2	IF Task is <i>turn</i> or <i>rotate</i> or <i>swing</i> THEN path is <i>circular</i> and NOT <i>linear</i> and <i>PTP</i> , and robot can execute the Task
3	IF Task is <i>bring</i> or <i>transfer</i> or <i>carry</i> THEN path is <i>PTP</i> and NOT <i>linear</i> and <i>circular</i> , and can execute the Task.
4	IF Task has <i>with tool</i> in its glossary, THEN robot CANNOT execute the Task.
5	IF Target has <i>slide</i> or <i>pull</i> or <i>push</i> in its glossary, THEN path is <i>linear</i> , and robot can manipulate the Target.
6	IF Target has <i>turn</i> or <i>rotate</i> or <i>swing</i> in its glossary, THEN path is <i>circular</i> . And robot can manipulate the Target.
7	IF Target has <i>bring</i> or <i>transfer</i> or <i>carry</i> in its glossary, THEN path is <i>PTP</i> , and robot can manipulate the Target.
8	IF Target has <i>hinge</i> as its part, THEN path is <i>circular</i> , and robot can manipulate the Target.

TABLE II. INITIAL PROBABILITIES AND CERTAINTIES

No.	$P_p(w)$	$C_p(w)$
1	$P_{lnr}(slide)=1.0$ $P_{lnr}(pull)=1.0$ $P_{lnr}(push)=1.0\dots$ $P_{imp}(slide)=0.0\dots$ (total 15pcs)	$C_{lnr}(slide)=1.0$ $C_{lnr}(pull)=1.0$ $C_{lnr}(push)=1.0\dots$ $C_{imp}(slide)=1.0\dots$ (total 15pcs)
4	$P_{imp}(bore)=1.0\dots$ (total 6pcs)	$C_{imp}(bore)=0.8\dots$ (total 6pcs)
6	$P_{ccr}(screw)=1.0\dots$ (total 460pcs)	$C_{ccr}(screw)=0.8\dots$ (total 460pcs)
8	$P_{ccr}(gate)=1.0\dots$ (total 6pcs)	$C_{ccr}(gate)=1.0\dots$ (total 6pcs)

Next, sets of $P_p(w)$ and $C_p(w)$ in TABLE II. are propagated to the others set having no values. The system starts a depth-first search from the root node. If it finds the node with value, it calculates $P_p(w)$ and $C_p(w)$ in adjoining nodes. The calculation method depends on the hierarchical relationship between current node and the adjoining nodes. The following is the basic principle for the method.

- Child inherits concepts of parent.
- Concepts of parent are weighted averages of concepts of children.
- Among brothers, only concepts that are inherited by parent are common. Unique concepts of brothers do not have correlations.

In order to use the value with the most reliable concept, the node(s) within only one hierarchy is used for the calculation even if nodes in different hierarchy have values. Its priority is parent > child > brother. Equations for the calculation are shown from (1) to (6); x represent calculated node, p , c , b represent parent, child and brother node, n in child and brother nodes is number of nodes.

using parent node

$$P_p(x) = P_p(p) \quad (1)$$

$$C_p(x) = C_p(p) \quad (2)$$

using child nodes

$$P_p(x) = \frac{\sum_{i=1}^n (P_p(c_i) * C_p(c_i))}{\sum_{i=1}^n C_p(c_i)} \quad (3)$$

$$C_p(x) = \frac{\sum_{i=1}^n C_p(c_i)}{n} \quad (4)$$

using brother nodes

$$P_p(x) = \frac{\sum_{i=1}^n (P_p(b_i) * C_p(b_i))}{\sum_{i=1}^n C_p(b_i)} \quad (5)$$

$$C_p(x) = 0.0 \quad (6)$$

To use $P_p(w)$ with high certainty, equation(3) and (5) apply weighted average with $C_p(w)$ as their weightings. (5) is executed after all brother nodes are searched, moreover, only sets of $P_p(b_i)$ and $C_p(b_i)$ with values are used for the calculation. When brother nodes are used parent node has no value, namely these values are not inherited by the parent. Therefore, $C_p(w)$ of (6) is 0.0.

After a value in a node is calculated, the value is used to calculation for its adjoining nodes without value. Therefore, the calculations propagate the values of TABLE II. like Figure 3. This propagation is continued until all nodes have values. To utilize not only parent but also child and brother and to retrace the hierarchy makes the system enable to set values in all nodes, even if small number of nodes receive the initial values. On the other hand, the system cannot calculate the values when no node receives initial value such as right upper tree in Figure 3.

D. Reasoning

The process of the reasoning is as follows: i. judgement whether task order is executable or not, ii. decision of a priority about Task or Target, iii. reasoning of path, iv. removal of discrepancies of the path between Task and Target.

In i., the system judges impossible task with two steps: before and during the reasoning. Before the reasoning, the system uses lexicographer files. The files stand for an index based on syntactic category and logical groupings, which are defined by WordNet. Specifically, in the case of Task, the word that does not belong to the lexicographer file “contact” or “motion” means inexecutable task. In the case of Target, a word that does not belong to the file “artifact” means inexecutable target. On the other hand, during reasoning the system considers $P_{imp}(w)$ and $C_{imp}(w)$ to detect an inexecutable task or target.

In ii., the system uses Task preferentially for the reasoning. It uses Target only when Task is without believability. We define “with believability” as $P_p(w) * C_p(w) \geq path_thr$, its value is 0.25 in this study. In addition, the system also applies Task if $P_p(w) * C_p(w) < path_thr$ in both Task and Target.

In iii., the system regards path(s) that has maximum $P_p(w) * C_p(w)$ and its proximity, which are calculated by equation(7) and (8) as answer. It judges “proximity” using

P_thr and C_thr . These values are 0.2 in this study. When all $P_p(w) * C_p(w)$ are 0.0, the system uses only $P_p(w)$.

$$p^* = \arg \max_p \{P_p(w) \cdot C_p(w)\} \quad (7)$$

$$p \in proximity \Leftrightarrow$$

$$\begin{cases} P_p(w) \cdot C_p(w) \geq mlt & P_{p_thr}(w) \geq 0 \cap C_{p_thr}(w) \geq 0 \\ P_p(w) + C_p(w) \geq add & \text{else if } add \geq 0 \\ P_p(w) + C_p(w) \geq 0 & \text{otherwise} \end{cases} \quad (8)$$

$$\text{where } P_{p_thr}(w) = P_{p^*}(w) - P_thr,$$

$$C_{p_thr}(w) = C_{p^*}(w) - C_thr,$$

$$mlt = P_{p_thr}(w) \cdot C_{p_thr}(w),$$

$$add = P_{p_thr}(w) + C_{p_thr}(w)$$

In iv., if the system uses Task for the reasoning and the number of reasoned paths are more than one, it cancels paths with $P_p(Target) * C_p(Target) < path_thr$. However, if all paths are cancelled, the canceling is invalid and all paths remain in the solution set. On the other hand, if the system uses Target, it cancels reasoned path(s) with $P_p(Task)=0.0$ and $C_p(Task)=1.0$.

IV. LEARNING

“Leaning” in this study stands for updating $P_p(w)$ and $C_p(w)$ for more precise reasoning. When the system displays reasoned path(s), it asks a user some items on the lower left part in output dialog in Figure 2. Each question decides the following: Q.1. suffix p in $P_p(w)$ and $C_p(w)$, Q.2. w (i.e. Task or Target), Q.3. and 4. value of $P_p(w)$, Q.5. value of $C_p(w)$. Q.1. is also used to confirm the framework. Since the system allows the user only one choice of p , the system learns only one kind of path per learning cycle even if the task order has two possibilities regarding the path. In order to a non-professional user can answer these questions easily, the system applies qualitative options and maps these options to numerical values.

Once the user gives a reply to the above questions, the system updates $P_p(w)$ and $C_p(w)$ in Task or Target and their neighbors’ node. During the update, it considers hierarchical relationship as in III.C. The update progresses under the following steps: i. current node, ii. all lower nodes of i., iii. brother nodes, iv. all lower nodes of iii., v. parent node. After v. has done, the system resets parent node to current node, and repeats from iii. to v. This makes the user’s reply propagates through hierarchical relationships. When the current node reaches the root node or is not updated, the system exits the propagation. The following is a procedure of the update in the each step.

First, the system calculates $P_p(w_{imp})$ and $C_p(w_{imp})$: temporary updating values, using $P_p(w_{user})$ and $C_p(w_{user})$: answer from a user, and $P_p(w_j)$ and $C_p(w_j)$: neighbor nodes of current node. Suffix j means before update, $j+1$ means after update. The equations to calculate $P_p(w_{imp})$ and $C_p(w_{imp})$ are changed depending on the steps as follows:

- In i., $P_p(w_{imp}) = P_p(w_{user}) * \alpha$ and $C_p(w_{imp}) = C_p(w_{user}) * \alpha$. Where α is an uncertainty factor. It is applied

to consider uncertainty of the reply from the user. Its value is 0.8 in this study.

- In ii. and iv., equation(1) and (2) are applied.
- In iii., equation(5) and (6) are applied.
- In v., equation(3) and (4) are applied. Here a node with $C_p(c_i)=0.0$ means it was calculated from brother nodes, hence we have no confidence in its $P_p(c_i)$. In calculation, the system skips such nodes and applies only those child nodes with $C_p(c_i)>0$ to prevent undesired decreases of $C_p(w_{imp})$. Additionally, the system makes sure that $C_p(c_i)$ that stem from production rules or user input remain >0 .

Next, The system calculates $P_p(w_{j+1})$ and $C_p(w_{j+1})$ using $C_p(w_{imp})$ and the history of propagation. The history of propagation means which hierarchy is used to calculate $P_p(w_j)$ and $C_p(w_j)$. There are three kinds of the calculations:

- If $C_p(w_{imp}) \gg C_p(w_j)$ or the hierarchy in the propagation history is the same as the hierarchy of the values used for calculating $C_p(w_{imp})$, then $P_p(w_{j+1})=P_p(w_{imp})$ and $C_p(w_{j+1})=C_p(w_{imp})$.
- If $C_p(w_{imp})$ and $C_p(w_j)$ are close, $P_p(w_{j+1})$ is calculated by weighted average of them. Where $C_p(w_{imp})$ and $C_p(w_j)$ are weightings. Also average between $C_p(w_{imp})$ and $C_p(w_j)$ becomes $C_p(w_{j+1})$. To judge the closeness the system reapplies C_{thr} used in III.C. If the absolute value of $C_p(w_{imp})-C_p(w_j)$ is within C_{thr} , it judges they are near.
- If $C_p(w_{imp}) \ll C_p(w_j)$, the system exits the calculation; it means end of propagation.

V. REASONING RESULTS

A. Results Using Only Production Rules

Figure 4. shows reasoning results of path(s) and their $P_p(w)*C_p(w)$ from six task orders. Where no values within the graph mean null, which are not calculated due to lack of corresponding production rules of TABLE I. in their trees. In the graph, $P_p(w)*C_p(w)$ corresponding to answers show high values. Although *open* includes various meanings and its values are null, the system reasoned efficiently using *drawer* or *door*. On the other hand, result

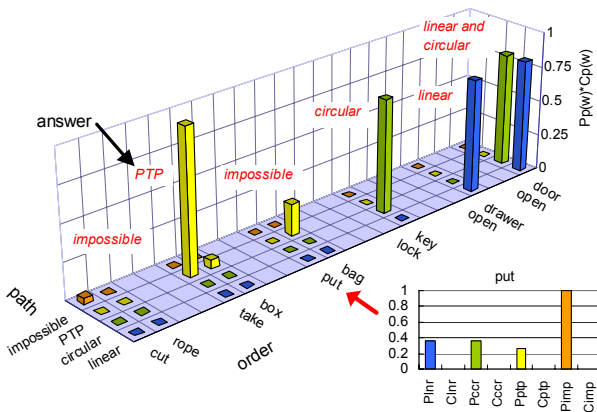


Figure 4. Reasoning results through only production rules

of *put the bag* is not *PTP* but *impossible*. This is due to the word *dibble* that is two ranks below from *put*. The rule No.4 from TABLE I. resulted in $P_{imp}(dibble)=1.00$ and $C_{imp}(dibble)=0.80$, and then, they were propagated to *put*. Even though the influence of *dibble* on *put* was weakened by the 135 child nodes of *put*, the value of $C_{imp}(put)$ became 0.0015 whereas all other C_p were 0.0000. In addition, $P_{pp}(bag)*C_{pp}(bag)$ is 0.22; it is under C_{thr} . The reason of this problem is shortage of reliable knowledge. Learning as shown in B. can solve it.

TABLE III. shows some results of *impossible* task judged through the lexicographer files. The system gives natural language explanations based on the lexicographer file name.

TABLE III. REASON OF IMPOSSIBLE TASK

Task	Target	Reason of Impossible
connect	Internet	I'm not clever enough to connect (connect belongs to lexi. file "communication").
persuade	wife	You can treat wife better than me(wife belongs to lexi. file "person").
pour	coffee	I can manipulate if coffee is in a container (coffee belongs to lexi. file "food, drink").
sew	skirt	sew is too complicated (sew belongs to lexi. file "creation").

B. Results After Learning

First, we evaluated effectiveness through one time learning. Figure 5. shows the result of "*brush the floor*". After the system learned "*wipe the table is linear*", it changed the result from *any kind of path* to *linear*, due to the propagation from $P_{lnr}(wipe)$ to $P_{lnr}(brush)$. On the other hand, a joint between *table* and *floor* is above four ranks; the propagation of $P_{lnr}(table)$ and $C_{lnr}(table)$ was stopped at the third node. The limit of propagation depends on the number of brother nodes and their value. A rough average of the limit is a few ranks above.

Next, we evaluated the percentage of right answers versus the number of leaning. TABLE IV. shows results of learning about typical domestic tasks. TABLE V. shows trained data that caused change of the results. The ratio of right answers was 0.3 at 0time, after it dropped to 0.2 it rose to 0.8 at 40times. At initial stage, many $C_p(w)$ were

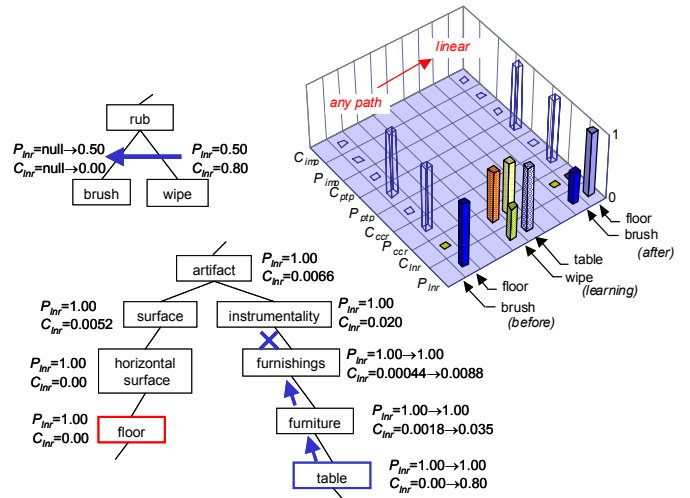


Figure 5. Effectiveness of one learning

null or 0.0 due to shortage of knowledge. Hence it could not decide unique path. Meanwhile, $P_{imp}(w)$ were also low; it answered *all kinds of path*. However, reliable $P_p(w)$ had been increased by acquisition of knowledge through the learning, hence, the correct ratio had also increased.

While the learning, Oscillation of $P_p(w)$ in words that include multiple meaning is a significant point. *Close* is typical example. In *close the door* or *close the drawer*, at learning No.0 the system used $P_p(door)$ and $C_p(door)$ and $P_p(drawer)$ and $C_p(drawer)$ for the reasoning due to null of $P_p(close)$ and $C_p(close)$, as well as *open the door* in Figure 4. However, after it learned *close=circular* at No.1, it reasoned *circular* using only $P_{ccr}(close)$ and $C_{ccr}(close)$ under the strategy in III.D. This reasoning continued up to No.30. After it acquired overall knowledge from the learning of *close=linear* at No.31, it was able to reason correctly. When independent $P_p(w)$ and $C_p(w)$ are applied, a learning of some path does not influence $P_p(w)$ and $C_p(w)$ of the others path. Therefore, after enough number of the learning the values of $P_p(w)$ and $C_p(w)$ are converged. In other word, the reasoning moves closer into a correct answer gradually with the learning.

TABLE IV. RESULTS OF LEARNING

Task	Target	Answer	Reasoning after learning				
			0*	10	20	30	40
close	door	L&C	L&C	C	C	C	L&C
close	drawer	L	L	C	C	C	L
fold	news-paper	C	P	P	P	P	P
lift	box	L	A	A	L	L	L
mop	corridor	L	A	A	A	L	L
open	refrigerator	C	A	A	A	A	A
return	book	P	P	P	P	P	P
turn off	printer	L	A	L	L	L	L
take away	dish	P	A	A	A	P	P
vacuum	carpet	L	A	A	L	L	L
Right answers ratio		1	0.3	0.2	0.4	0.6	0.8

*: number of learning, L:Linear, C:Circular, P:PTP, A: All kinds of path

TABLE V. BENEFICIAL TRAINING DATA

No.	Task	Target	Trained $P_p(w)$ *	Effect
1	close	cap	$P_{ccr}(close)=0.75$ $P_{ccr}(cap)=0.75$	$P_{ccr}(close)=null \rightarrow 0.75$ $C_{ccr}(close)=null \rightarrow 0.80$
6	switch off	stereo	$P_{mr}(switch\ off)=0.50$ $P_{mr}(stereo)=0.75$	$P_{mr}(turn\ off)=0.36 \rightarrow 0.50$ $C_{mr}(turn\ off)=0.00 \rightarrow 0.80$
18	brush	suit	$P_{mr}(brush)=0.75$	$P_{mr}(vacuum)=null \rightarrow 0.75$ $C_{mr}(vacuum)=null \rightarrow 0.00$
19	pick up	table	$P_{mr}(pick\ up)=1.00$	$P_{mr}(lift)=0.33 \rightarrow 1.00$ $C_{mr}(lift)=0.00 \rightarrow 0.02$
25	swab	floor	$P_{mr}(swab)=0.75$ $P_{mr}(floor)=1.00$	$P_{mr}(mop)=null \rightarrow 0.75$ $C_{mr}(mop)=null \rightarrow 0.26$
27	bring out	dress	$P_{pp}(bring\ out)=1.00$	$P_{ccr}(take\ away)=null \rightarrow 1.00$ $C_{ccr}(take\ away)=null \rightarrow 0.00$
31	close	shutter	$P_{mr}(close)=0.50$ $P_{mr}(shutter)=1.00$	$P_{mr}(close)=1.00 \rightarrow 0.50$ $C_{mr}(close)=0.11 \rightarrow 0.80$

*: All $C_p(w)=0.8$

C. Limitation

There are three problems. First, the system cannot apply tasks whose path is not well defined such as *rinse the dish*. To solve it, the user needs to teach *impossible* and use more path-dependent task like *take away the dish* and *switch on the dishwasher*. Second, as shown in *pour the coffee* in TABLE III, the reasoning of some task that needs equipment to manipulate Target is often incorrect. In other words, the system cannot reason what a robot manipulates to realize ordered task. To solve this problem, the user needs to change task order to more equipment-dependent form like *tilt the pot*, or needs to reply path of a contact point between an equipment and an end effector instead of CPT. Three, In the learning it is difficult to reply exact $P_p(w)$ and $C_p(w)$ because the user figures out replies by his/her instinct. One solution is learning the tendency of the instinct by Bayesian Network and their reflection to α .

VI. CONCLUSION

This paper proposed a method for reasoning intent of motion: a brief path, through a task order. Each word in an electric thesaurus is set combinations of $P_p(w)$: probability of *linear*, *circular* or *PTP* path or *impossible*, and $C_p(w)$: certainty of $P_p(w)$. After setting $P_p(w)$ and $C_p(w)$ using 8 production rules and hierarchical relationship of words, path(s) has high $P_p(w)*C_p(w)$ in words included in the task order becomes answer. Moreover, update and re-propagation of $P_p(w)$ and $C_p(w)$ using answers from a user enable more precise reasoning. Right answers ratio is increased from 0.3 to 0.8 by 40times learning.

This paper focused on path since it is the most distinguishing feature of task. Now, we address reasoning of detailed items, such as the speed of the target and the grasp force, using the same algorithm. In parallel, we further develop the proposed teaching system. When Semantic Web will be ready, we will apply it. Importing both basic concepts and unique data of targets such as shape and usage will enable in-depth and precise reasoning. Its form will be more natural and closer to human.

REFERENCES

- [1] Y. Kuniyoshi, M. Inaba and H. Inoue: "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," IEEE Trans. Robotics & Autom, Vol.10, no.6, pp. 799-822, 1994.
- [2] Buchanan B.G. and E.A.Feigenbaum: "DENDRAL and Meta-DENDRAL: Their applications dimension," Artificial Intelligence 11, pp.5-24, 1978.
- [3] Pople.H.E, Jr., J.D.Myers and R.A.Miller:"DIAROG: A Model of diagnostic logic for internal medicine," IJCAI 4, pp.848-855, 1975.
- [4] U. Ahlrichs, J. Fischer, J. Denzler, Ch. Drexler, H. Niemann, E. Nöth, and D. Paulus: "Knowledge Based Image and Speech Analysis for Service Robots," Proc. IEEE Integration of Speech and Image Understanding, pp. 21-47, 1999.
- [5] Wermter S., Elshaw M., Farrand S.:"A Modular Approach to Self-organisation of Robot Control Based on Language Instruction," Connection Science, 2003.
- [6] <http://www.w3.org/2001/sw/>
- [7] <http://www.w3.org/TR/2001/NOTE-dam1+oil-walkthru-20011218/>
- [8] <http://www.cogsci.princeton.edu/~wn/>
- [9] Shortliffe, E.H.:"Computer-based medical consultation: MYCIN," New York: American Elsevier, 1976.