A Unified Linguistic Formalism for Teleoperation of Semi-Autonomous Robot*

Sungmoon Joo, *Member, IEEE*

Abstract— **The ultimate goal of our work is to develop a novel, integrated system for semi-autonomous robotic teleoperation by introducing an interfacing language shared by human operators and robots. This work presents a unified linguistic formalism based on the combinatory categorial grammar (CCG) and preliminary result towards the goal; we show that, using a language derived from a CCG, human-to-robot communication can be modeled as a hierarchical task network (HTN) planning problem, and robot-to-human communication as a plan recognition problem.**

I. Introduction

The interest in the application of robots in the search and rescue missions, underwater or space explorations, and battlefield missions has significantly increased over the last two decades [4,12]. Fully autonomous robot operation would be ideal in these applications. However it is still quite challenging for robots to perceive, reason, and perform actions autonomously in unstructured and dynamic environments. As an alternative, teleoperating semi-autonomous robots is considered more realistic and viable, evidenced by many field robots in service [12]. The major technical challenges envisioned by these robotic teleoperation are facilitating the supervision of robots by operators who do not have expertise in robotics, lowering the operator's workload, and the effective teleoperation with low-bandwidth, high latency, and intermittent communications [10-12].

Given a shared language between human operators and robots, robots could inform operators the task progression, what they expect they could do, and humans could request the robot perform tasks or course of actions. We believe this collaborative, semi-autonomous teleoperation via low-bandwidth communication between humans and robots to yield significant improvement over the direct, non-collaborative teleoperation, and to provide a solution to most of the challenges mentioned earlier.

The ultimate goal of our work is to develop a novel, integrated system for collaborative, semi-autonomous teleoperation by introducing an interfacing language shared by human operators and robots. This work presents a preliminary work towards to the goal. The main contributions of this work are: (i) We present a unified linguistic formalism for efficient human-robot interaction (HRI), based on a lexicalized grammar formalism called combinatory categorial grammar (CCG), (ii) We provides a new perspective for HRI in the context of planning and plan recognition

The remainder of this paper is organized as follows: related work is reviewed in Section II. We provide intuitions behind this work, and introduce the proposed system concept in Section III. In Section IV, an informal definition of CCG is given. Next, in Section V, we provide a unified linguistic formalism based on CCG to model (i)human-robot communication in teleoperation, in the context of planning plan recognition, and (ii)affordance-based object-action schema.. Finally, we conclude in Section VIII with summary, discussion and future work.

II. Related Work

Application of techniques from linguistic theory to robotics is not new [8,9,13,14,17-19,21-27]. Artificial intelligence (AI) community and linguistic theory community have been cooperatively developing their research fields. In particular, as interest in HRI has grown, natural language processing became very active area of research [4]. Part of this work is heavily influenced and motivated by studies on linguistic plan recognition [5,6], and spoken language processing for HRI [21-27].

Many systems have exploited the compositional structure of language to statically generate a plan corresponding to a natural language command [9,18]. Our work moves beyond this framework by providing a unified linguistic formalism for teleoperation of semi-autonomous robots.

Our approach tightly combines high-level action planning and motion generation by providing new affordance-based object-action schema under the unified linguistic formalism. Others [18,21-27] have used generative and discriminative models for understanding human commands in natural language, but did not fully exploit the hierarchical nature of the linguistic structure and exploit the analogies between HTN planning framework. Some previous work focuses on the usage of instructions for tasks available on the web, and inferring about whether the robot has the capabilities for the instructions, and those systems usually need separate low-level language for execution of the high-level action plan [9,18], while we are providing a linguistic framework. This paper reports preliminary result on combining previous work in planning, plan recognition, natural language processing, and spoken-language-based HIR in robotics with a unified linguistic formalism. To authors' best knowledge, our approach is novel in the sense that we provide a unified, CCG-based linguistic formalism for teleoperation of semi-autonomous robots.

*Research supported by the Defense Advanced Research Project Agency (DARPA) award D13AP00047.

[1]Sungmoon Joo is a research scientist in the School of Interactive Computing, and with the Institute for Robotics and Intelligent Machines (IRIM), Georgia Institute of Technology, Atlanta, GA 30332 USA (phone: 404-834-1491; fax: 404-385-8558; e-mail: sungmoon.joo@cc.gatech.edu).

III. Concept of The Proposed System

Communication between humans and robots would be one of four types: Directives, Queries, Assertives, and Correctives [17-19]. In this work, we concentrate on the directives, as we are interested in the generation of task-oriented action sequences (planning), and monitoring of robot's task execution (plan recognition) in robot teleoperation.

Since communication with natural language is convenient for humans, many studies have proposed schemes for HRI based on natural language [4]. Although natural language is clearly an effective medium for human-human communication, it presents several problems when used for human-robot communications; Natural language descriptions tend to be underspecified, diverse, vague, or ambiguous for robot to understand [4,23,25-27]. Thus restricting natural language to a small finite set of linguistic expressions (i.e. a formal language) would mitigate these problems.

A. Linguistic Planning and Plan Recognition

Consider the following directive task description as an example. This example is motivated by one of the tasks in the DARPA Robotics Challenge [14], and will be used throughout the paper. For simplicity, we concentrate only core parts of the sentence (i.e. ignoring inflections, etc.)

Robot pick up the wooden block from the ground

We, humans, can easily rephrase the sentence as a compound sentence that consists of four sub-sentences representing the subtasks that must be performed for the original task:

Sub-sentence #1: Robot move to near the block (if necessary).

Sub-sentence #2: Then, robot search the grasp point.

Sub-sentence #3: Then, robot grasp the block.

Sub-sentence #4: Then, robot lift up the block.

Sub-sentence 3 can also be further decomposed to clarify how associated subtasks are performed:

Sub-sentence #3-1: Robot move (right) hand to the block.

Sub-sentence #3-2: Then, robot close fingers.

This decomposition process continues until all of the tasks are represented by primitive tasks that are not decomposable further. In the example above, a set of primitive tasks and the command pairs, O, can be defined as follows:

O = {(move to, GO_TO), (search, SEARCH), (move hand to, MOVE_HAND_TO), (close fingers, CLOSE_FINGERS), (lift up, MOVE_HAND_UP)}.

Finally, the original task description can be executed by instantiating a sequence of actions with associated parameters):

GO_TO;SEARCH;MOVE_HAND_TO;CLOSE_FINGERS;MOVE_HAND_UP

where ; is the sequence operator from the dynamic logic [15,23].

This sequence of primitive tasks is essentially a plan for robot to perform the given (high-level or goal) task. Note that the planning problem is easily solved by a proper hierarchical task decomposition based on human input (i.e. domain knowledge on how to decompose the task), and the task decomposition may not be unique.

Figure 1 visualizes the resulting task decomposition.

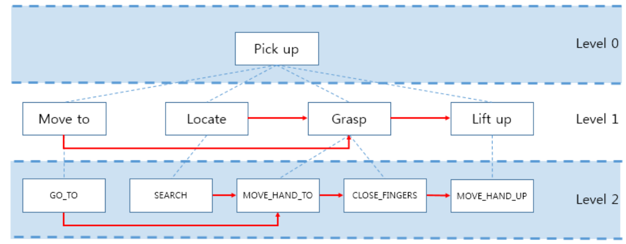

Figure 1. Hierarchical Task Decomposition.

The suitability of a given hierarchical task description depends on the level of granularity required. In the example, non-expert operators will most likely prefer describing tasks at level 0 as they might not have detailed knowledge on the primitive tasks. On the other hand, an expert operator may prefer specifying a given task step by step or reconfiguring its subtasks according to various circumstances; in such cases, interacting with robots at level 1 or level 2 is desirable. Thus, the hierarchy allows operators to interact with robots at various levels of task hierarchy. The hierarchical task decomposition potentially enhances the expressive power of our task description by building up its vocabulary while ensuring the interpretability of task as well as the interactivity by allowing human-robot interactions to take place at various levels of the hierarchy.

Inferring on the current task progress or situation can be modeled as an inverse process of planning. This problem is known as plan recognition [5,6]. If a robot is aware of the task hierarchy as given in Fig. 1, and observes the execution of primitive tasks in sequence, the robot can make inference on the sub-tasks and even the original task by properly parsing the observation. For example, if the observation history at time t, is given by

$h_{[1:t]}$ = (GO_TO, SEARCH, MOVE_HAND_TO, CLOSE_FINGERS),

the robot could report to the human operator that it just completed 'Pick' task, and, infer that the remaining sub-task to accomplish the goal task is 'Lift up'

B. Teleoperation of Semi-Autonomous Robot

We assume the robot is a model and task-based rational agent with limited capability; If the robot is (fully) autonomous, it should be able to keep track of the world state as well as a set of tasks/goals it is trying to achieve, chose an action that will lead to the achievement of its goals, and executes the action on its own [3]. Semi-autonomous robots, however, rely on inputs from human operators to some extent. In this work, we concentrate on the task-level communication between the human operator and the robot; a clear division of labor (between human operator and robot) is left for the future work, as it is another area of active research [4].

Fig. 2 is a conceptual description of the system we propose. The key idea behind the task-level communication is linguistic interpretation of planning; The core of the proposed teleoperation system has three components: translator, planner and plan recognizer (shaded boxes in Fig. 2). The translator parses operator's commands (e.g. spoken natural language input, command line input) into a (shared) formal language. If the command is high-level, the planner decomposes into low-level, primitive actions so that the robot can executes. This high-level communication is particularly beneficial under low-bandwidth communication. We consider two main channels for robot-to-human communication: direct and indirect observation. The robot interprets the observation and "explains" via indirect, high-level communication channel. The "explanation" is close enough to natural language (i.e. structured/controlled natural language) so that human operators can understand without additional translation, and is preferred over chatty communication through the direct observation channel when the bandwidth is limited.

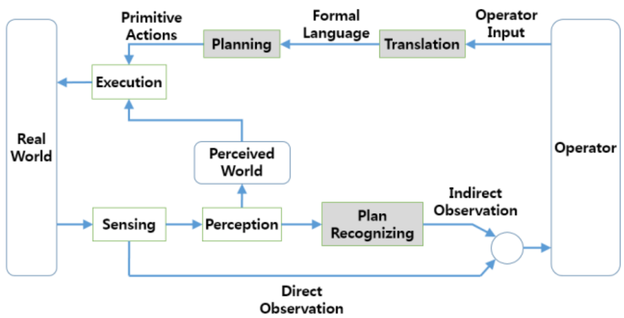

Figure 2. Teleoperation of Semi-Autonomous Robot

IV. Combinatory Categorial Grammar

In this work, we will argue for building structures for the shared (translated, formal) language based on a particular lexicalized grammar formalism called the combinatory categorial grammar (CCG). This section serves as a brief introduction to CCG, summarizing previous work [5,6,15].

In CCG, elements like verbs are associated with a syntactic "category" which identifies them as functions, and specifies the type and directionality of their arguments and the type of their results.CCG uses a small set of combinatory rules (i.e. combinators) to combine rich lexical categories of 'words'. Categories in CCG are either atomic or complex:

(i) Atomic categories: A finite set of basic action categories C (i.e. $A, B, \cdots \in C$). These categories correspond to propositions that describe world states that result from executing actions (e.g. grasping the block).

(ii) Complex categories: Complex categories are functions that take a set of arguments$\{W, X, ...\} \subset C$ and produce a result $Z \in C$.

For example, the complex category $Z \backslash \{W, X, ...\}$ is a functor category that takes an unordered set of arguments $\{W, X, ...\} \subset C$ and produces a result Z where the slash indicates where the function looks for its arguments. Complex categories specify the type and direction of the arguments and the type of the result. Here, we use the "result leftmost" notation in which a rightward-combining functor over a domain B into a range A are written A/B, and leftward-combining functor is written A\B.

To parse sentences a number of combinators are required:

•Forward & Backward application:$\dfrac{A/B \;\; B}{A} >$, $\dfrac{B \;\; A \backslash B}{A} <$

•Forward & Backward composition:

$$\frac{A/B \;\; B/C}{A/C} > \mathbf{B} \, , \frac{B \backslash C \;\; A \backslash B}{A \backslash C} < \mathbf{B}$$

CCG also allows one category to be "type-raised" into another category. For example,

•Forward & backward type raising:

$$\frac{A}{B/(B \backslash A)} > \mathbf{T} \, , \frac{A}{B \backslash (B/A)} < \mathbf{T}$$

Steedman[15] noticed that combinators that contribute to the additional power of CCG(over traditional context-free grammars, CFGs), such as composition and type-raising, play key roles in modeling affordances and action-perception schemas.

An instance of a CCG combinatory is obtained by substituting CCG categories for variables. To parse a sentence is to apply instances of CCG combinators so that the final category is derived at the end. For example,

$$\cfrac{\cfrac{\cfrac{the}{NP/N} \quad \cfrac{robot}{N}}{NP} > \quad \cfrac{grasp}{(S \backslash NP)/NP} \quad \cfrac{\cfrac{the}{NP/N} \quad \cfrac{block}{N}}{NP} >}{S} < \quad (1)$$

V. CCG-based Unified Linguistic Formalism

In this section, we will show how a language derived from a CCG (i.e. strings of terminal symbols defined in a CCG) can be used for both planning and plan recognition more formally. An action language included in the formalism is a STRIPS-style action description language [3] represented as primitive tasks (i.e. operators) in the Hierarchical Task Network (HTN). Let's revisit the example considered in Section I (Figure 1). To achieve the goal task "Pick up" the robot must perform a sequence of actions GO_TO, SEARCH, MOVE_HAND_TO, CLOSE_FINGERS, and MOVE_HAND_UP. Note that GO_TO and SEARCH must be executed before MOVE_HAND_TO but are unordered with respect to each other. We consider plan recognition first to help readers understand the concept of CCG, and how CCG is used in the context of planning and plan recognition.

A. Plan Recognition using CCG

One possible definition of the lexicon for the example is as follows:

•GO_TO : Move to

•SEARCH : Search

•MOVE_HAND_TO : PreGrasp\{Move to,Search}

•CLOSE_FINGERS : Grasp\PreGrasp

•MOVE_HAND_UP : Pick up\Grasp

To see how combinators introduced earlier combine with the lexicon to allow for parsing the observation of a sequence of primitive actions to recognize the high level goals, consider the following derivation:

Step 1. $\dfrac{\dfrac{GO\_TO}{Move\ to} \quad \dfrac{SEARCH}{Search} \quad \dfrac{MOVE\_HAND\_TO}{PreGrasp\backslash\{Move\ to, Search\}}}{PreGrasp} <$

Step 2. $\dfrac{\dfrac{GO\_TO\ SEARCH\ MOVE\_HAND\_TO}{PreGrasp} < \dfrac{CLOSE\_FINGERS}{Grasp\backslash PreGrasp}}{Grasp} <$

Step 3. $\dfrac{\dfrac{GO\_TO\ SEARCH\ MOVE\_HAND\_TO\ CLOSE\_FINGERS}{Grasp} < \dfrac{MOVE\_HAND\_UP}{Pick\ up\backslash Grasp}}{Pick\ up} <$

The derivation explains the goal task "Pick up" is accomplished by performing sub-tasks categoris "PreGrasp" and "Grasp" in sequence.

B. *Planning using CCG: Hierarchical Task Network*

It is very natural to represent the hierarchical task structure from the previous section as a Hierarchical Task Network (HTN) [3,7]. Here we adopt a set-theoretic HTN formalism [7], and introduce the simple task network with partial-order forward decomposition [3]. At first, we define the operator and the task:

Definition 1. Operator

An operator is a tuple:

$\gamma = (prec(\gamma), add(\gamma), del(\gamma)) \in 2^L \times 2^L \times 2^L$ for each $\gamma \in O$, where L is a finite set of proposition symbols, $prec$ is a precondition, add is an add list, del is a delete list, and O is a finite set of operators.

Definition 2. Task

A task is a specific activity that may be undertaken in the world. If a task is equivalent to a primitive action (i.e. operator), then the task is primitive. Otherwise, it is non-primitive (or compound).

The task hierarchy can be represented using the concept of task network. A task network is a set of tasks that need to be performed, along with constraints on the tasks.

Definition 3. Task Network

A task network is a partially-ordered multiset of task names. A task network is a tuple $tn = (T, \prec, \alpha)$ where

• T is a finite nonempty set of task symbols.

• $\prec \ \subseteq T \times T$ is a partial order over T.

• $\alpha : T \rightarrow X$ is a map from T to a set of task names X

Tasks symbols serve the purpose of unique identifiers because task names may occur multiple times in the same task network. To expand the task network, we need to know how each task is decomposed into its' sub-tasks. This task decomposition is represented by the concept of "method":

Definition 4. Method

A method is a pair $m_{\alpha(t)} = (\alpha(t), tn_{\alpha(t)})$ where $t \in T$ is a task, $tn_{\alpha(t)}$ is a task network associated with the task t, and $\alpha$is a mapping that maps a task to its' name.

Definition 5. Task Decomposition

A method $m = (\alpha(t), (T_{\alpha(t)}, \prec_{\alpha(t)}, \alpha_{\alpha(t)}))$ decomposes a task network $tn_1 = (T_1, \prec_1, \alpha_1)$ into a new task networktn_2. The decomposition process is, written $tn_1 \xrightarrow{\text{t,m}} tn_2$, given by:

$$T' := T_1 \setminus \{t\}\, T_2 := T_1^{'} \cup T_{\alpha(t)}$$

$$\prec_1^{'} := \{(t_1, t_2) \in \prec_1 | t_1 \neq t \wedge t_2 \neq t\}$$

$$\begin{aligned} \prec_2 := & \prec_1^{'} \cup \prec_{\alpha(t)} \\ & \cup \{(t_1, t_2) \in T_1^{'} \times T_2 \mid (t_1, t) \in \prec_1\} \\ & \cup \{(t_2, t_1) \in T_2 \times T_1^{'} \mid (t, t_1) \in \prec_1\} \end{aligned}$$

$$\alpha_2(t') := \begin{cases} \alpha_1(t') & \text{if } t' \in T_1^1 \\ \alpha_m(t') & \text{if } t' \in T_{\alpha(t)} \end{cases}$$

$$tn_2 := (T_2, \prec_2, \alpha_2)$$

If there is a finite sequence of task decompositions $tn_1 \rightarrow_D tn_2 \rightarrow_D \cdots \rightarrow_D tn_n$, then we write $tn_1 \rightarrow_D^* tn_n$.

Now we can formally define a HTN planning domain and the associated HTN planning problem:

Definition 6. HTN planning domain

An HTN planning domain is a tuple $D_{HTN} = (S, C, O, M, \gamma)$, where:

• S is a finite set of robot states (represented by propositional symbols).

• C is a finite set of compound task names.

• O is a finite set of primitive task names.

• $M \subset C \times TN_{C \cup O}$ is a set of methods over task names.

• $\gamma : S \times O \rightarrow S$ is a partial function for state transitions. If $\gamma(s, a)$ is not defined, than $a \in O$ is not applicable in $s \in S$.

Definition 7. HTN planning problem

An HTN planning problem is a tuple (D, s_0, tn), where $D = (S, C, O, M, \gamma)$ is an HTN domain, $s_0 \in S$ is an initial state, and tn is a task network over the task names in D.

A task network is executable in a state s_0 for domain D if task network tn is primitive and there exists some total ordering over the tasks $t_1, \ldots, t_n$ in tn and a sequence of state $s_1, \ldots, s_n$ such that

$$\gamma(s_i, \alpha(t_{i+1})) = s_{i+1}, for\ 0 \leq i \leq n$$

An HTN planning problem is solvable if there is a sequence of decompositions $tn \rightarrow_D^* tn'$ such that tn' is executable in s_0.

For the example we have, one possible definition of the HTN domain is as follows,

• $C = \{Pick\ up, Grasp\}$ is a finite set of compound task names.

• $O = \{GO\_TO, SEARCH, MOVE\_HAND\_TO, CLOSE\_$ $FINGRERS\ MOVE\_HAND\_UP\}$ is a finite set of primitive task names, with $C \cap O = \emptyset$.

• $T = \{Goal, Move\ to, Search, Grasp', Lift\ up, Reach\ out,$ $Close\ fingers\}$ is a finite set of task symbols

• $\alpha = \{(Goal, Pick\ up), (Move\ to, GO\_TO), (Grasp', Grasp),$ $(Search, SEARCH), (Reach\ out, MOVE\_HAND\_TO),$ $(Close\ fingers, CLOSE\_FINGERS), (Lift\ up, MOVE\_HAND\_UP)\}$ is the labeling mapping.

• Task decomposition methods,

$$\begin{aligned} m_{Pickup} = &(Pick\ up, \{Move\ to, Search, Grasp, Lift\ up\}, \\ &\{(Move\ to \prec Grasp), (Search \prec Grasp), \\ &(Grasp \prec Lift\ up)\}, \{(Move\ to, GO\_TO), \\ &(Search, SEARCH), (Grasp', Grasp), \\ &(Lift\ up, MOVE\_HAND\_UP)\}) \\ m_{Grasp} = &(Grasp, \{Reach\ out, Close\ fingers\}, \\ &\{(Reach\ out \prec Close\ fingers)\}, \\ &\{(Reach\ out, MOVE\_HAND\_TO), \\ &(Close\ fingers, CLOSE\_FINGERS)\}) \end{aligned}$$

Note that how the partial ordering of 'Move to' and 'Locate' with respect to 'MOVE_HAND_TO' is represented by using $\prec$ operator, which was represented by using a category of type $A\backslash\{B, C\}$ in plan recognition.

Formally, the vocabulary of HTN language $\mathcal{L}$ is a tuple $\langle S_V, S_C, S_P, T_O, T_C, N\rangle$ where $S_V = \{v1, v2, \ldots\}$ is an infinite set of variable symbols, S_C is a finite set of constant symbols, S_P is a finite set of predicate symbols, T_O is a finite set of primitive task symbols, T_C is a finite set of compound task symbols, and $N = \{n_1, n_2, \ldots\}$ is an infinite set of symbols used for labeling tasks (i.e. task names).

C. Operator Command Translation with Semantic Mapping

CCGs model the semantics (meaning) of operator's commands as well as the syntax. Once operator's command is parsed into a formal language, we assign meanings to the parsed lexical items. The formal language we adopt in this work is a dynamic logic which is a subset of first-order dynamic logic (FDL) as proposed in [23]. This dynamic logic is also used to define the affordance in the next subsection. The semantics is represented by λ -expressions [15,23]. With λ-expressions and associated lexicon, we can translate (i.e. parse) operator's commands into a formal language that robot can understand. For more details, readers are referred to [23].

A λ-expression is composed of variables (e.g. x, y), the abstraction symbols lambda 'λ' and dot '.', and parenthesis ( ). The abstraction operator, λ , is said to bind its variable when it occurs in the body of the abstraction. Such variables are called bound variables. All other variables are called free variables. An expression that contains only bound variables is said to be closed. Closed λ-expressions are called combinators. For example, a λ abstraction $\lambda x.t$ represents taking a single input x and substituting it into the expression t. An application ts represents the act of calling function t on input s to produce $t(s)$. A β-reduction states that an application of the form $(\lambda x.t)s$ reduces to the term $t[x := s]$. The symbol @ can be used in the β -reduction (e.g. $(\lambda x.grasp(x))block\#1$ $= \lambda x.grasp(x)@block\#1$ to represent the function-argument relation more clearly [23]. For example, Table I shows an example of a formal language to translate operator command "Pick up the block".

TABLE I. LANGUAGE TRANSLATION TABLE

Operator language	Formal language: Semantics
Move to	GO_TO: $\lambda x.GO\_TO(x)$
Search	SEARCH: $\lambda x.SEARCH(x)$
Reach out	MOVE_HAND_TO: $\lambda x.MOVE\_HAND\_TO(x)$
Close fingers	CLOSE_FINGERS: $\lambda x.x@CLOSE\_FINGERS$
Lift up	MOVE_HAND_UP: $\lambda x.MOVE\_HAND\_UP(x)$
the block	$\lambda x.x@block$

D. Object & Action Representation: Affordance

In robotic teleoperation, the robot interact with objects in the world, through actions. In this section, we provide affordance-based object-action schema, and show that we can represent the schema using CCG [15]. This object-action schema increases robot's inference power for autonomy. Here we follow Steedman's work [15] on formalizing affordance using CCG.

Steedman [15] defined affordance as a relation between an object and an action. If we consider the block-pick-up scenario, for example, the relation between block (object) and lift-up (action) can be represented as follows,

$$\{o\} \wedge on\_the\_ground \multimap [a]in\_the\_air \Rightarrow$$
$$\{block\} \wedge on\_the\_ground \multimap [lift]in\_the\_air,$$

Where $\Rightarrow$ represents the standard or intuitionistic implication and $\multimap$ represents the linear logical implication indicating the change of the value of facts [15]. Actions are represented as functions using the lambda calculus and the linear logical implication. For example,

$$lift(block) = \lambda block.\{on\_the\_ground \multimap in\_the\_air\}$$

Now we can define an object in terms of the affordances of the object; The set of actions associated with an object constitutes the affordances of block. For example, if we want a robot to perform three actions – reach out, close fingers, lift_up - with a block, then the affordance of a block can be defined(Steedman's definition) as follows,

$$affordances(block) = \begin{Bmatrix} reach\ out \\ close\ fingers \\ lift\_up \end{Bmatrix}$$

Then, affordance-based object schema can be interpreted as a function which maps a block (object, o) into functions from their affordances (i.e. action a) to their results (state s). For example, the block schema can be represented as follows,

$$block' = \lambda s_{block}.\lambda a_{affordances(block)}.as$$

The operation of turning an object of a given type into a function over those functions that apply to objects of that type can be represented by the type-raising ($\mathbf{T}$) combinatory in CCG:

$$\mathbf{T}s \equiv \lambda a.as$$

Then the block schema can be rewritten:

$$block' = \lambda s_{block}.\mathbf{T}s_{block}$$

VI. Conclusions and Future Work

This work presents a unified linguistic formalism for teleoperation of semi-autonomous robots based on CCG, which facilitates human-robot communications and robot autonomy. The hierarchical nature of human directives naturally leads us to a new perspective for human-robot communication in the context of HTN planning and plan recognition. The choice of CCG formalism allows us to adopt and apply an affordance-based object-action schema theory developed in linguistic community, and to incrementally process human-robot communication. If we fully exploit (mildly) context-sensitiveness of CCG, the formalism proposed in this work could provide means for incorporating rich context from human environment and generating human-preferred motions [15,16].

On the other hand, we have carefully deferred many questions on potential issues: semantic ambiguities, disfluencies, under specifications, groundings, verifications just to name a few [17,18,21-27]. We have plans to extend the affordance-based object-action schema to better serve robot motion generation, and to apply the linguistic framework to low-level control to secure safe, real-time execution of commands [8,21]. Our near term goals include (i)to perform feasibility test of the proposed approach by building an on-line pipeline from operator command to robot action execution, and (ii)to migrate the system into our in-house, open-source articulated robot simulation environment, DART and Grip [20] for extensive simulation tests and the development of graphical user interface for teleoperation.

Acknowledgment

The author wish to thank Dr. Mike Stilman for motivating and supporting this work.

References

[1] A. V. Aho and J. D. Ullman, Foundations of Computer Science, New York, NY:W.H. Freeman/Computer Science Press, 1992.

[2] K. Erol, J. Hendler, and D. S. nau, "HTN Planing: Complexity and Expressivity," in Proceedings of the 12th National Conference on Artificial Intelligence (AAAI), pp. 1123–1128, 1994.

[3] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd edition, Prentice Hall, 2010.

[4] M. A. Goodrich and A. C. Schultz, "Human-Robot Interaction: A Survey," in Foundations and Trends in Human-Computer Interaction, Vol. 1, No. 3, pp. 203-275, 2007.

[5] C. W. Geib, "Using Lexicalized Grammars and Headedness for Approximate Plan Recognition," in Proceedings of the AAAI Workshop on Plan, Activity, and Intent Recognition, July, 2007.

[6] C. W. Geib and M. Steedman, "On Natural Language Processing and Plan Recognition," in Proceedings of International Joint Conferences on Artificial Intelligence, pp. 1612-1617, 2007.

[7] R. Alford, V. Shivashankar, U. Kuter, and D. Nau, " HTN Problem Spaces: Structure, Algorithms, Termination," in Proceedings of the 5th Annual Symposium on Combinatorial Search, 2012.

[8] N. T. Dantam and M. Stilman, "The Motion Grammar: Analysis of a Linguistic Method for Robot Control," IEEE/RAS Transactions on Robotics, Vol. 29, No. 3, pp. 704-718, 2013

[9] M. Tenorth, A. Perzylo, R. Lafrenz, and M. Beetz, "The RoboEarth language: Representing and Exchanging Knowledge about Actions, Objects, and Environments," in Proceedings of *IEEE International Conference on Robotics and Automation, May, 2012*.

[10] T. B. Sheridan, Telerobotics, automation, and Human Supervisory Control. The MIT Press, 1992.

[11] B. Willaert, J. Bohg, H. V. Brussel, and G. Niemeyer, "Towards Multi-DOF Model Mediated Teleoperation: Using Vision to Augment Feedback," 2012 IEEE International Workshop on Haptic Audio Visual Environments and Games, pp. 25-31, Munich, Germany, Oct. 8–9, 2012

[12] http://www.theroboticschallenge.org/aboutprogram.aspx

[13] T. Nakama, E. Munoz, K. LeBlanc, and E. Ruspini, "Facilitating Huma-Robot Interaction: A Formal Logic for Task Description", ROBOT2013: First Iberian Robotics Conference (Advances in Robotics, Vol. 2), Advances in Intelligent Systems and Computing Series, Vol. 253, pp. 331-344, Springer International Publishing, 2014.

[14] M. Tenorth and M. Beetz, "A Unified Representation for Reasoning about Robot Actions, Processes, and their Effects on Objects," in Proceedings of *IEEE/RJS International Conference on Intelligent Robots and Systems*, pp. 1351-1358, Vilamoura, Portugal, Oct., 2012.

[15] M. Steedman, "Plans, Affordances, and Combinatory Grammar," Linguistics and Philosophy, 25:723-753, 2002

[16] A. Joshi, K. Vijay-Shanker, and D. Weir, "The Convergence of Mildly Context-Sensitive Grammar Formalisms," Foundational Issues in Natural Language Processing, Cambridge, MA: MIT Press, pp.31-81, 1991.

[17] K. Hsiao, S. Tellex, S. Vosoughi, R. Kubat, D. Roy, "Object Schemas for Grounding Language in a Responsive Robot," Connection Science 20(4), pp.253-276, 2008.

[18] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller and N. Roy, "Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation," Proceedings of AAAI Conference on Artificial Intelligence, pp. 1507, 2011

[19] M. Bollini, S. Tellex, T. Thompson, N. Roy, and D. Rus, "Interpreting and Executing Recipes with a Cooking Robot," International Symposium on Experimental Robotics, Quebec City, Canada, 2012

[20] https://github.com /golems/grip2

[21] C. Matuszek, E. Herbst, L. Zettlemoyer, D. Fox, "Learning to Parse Natural Language Commands to a Robot Control System," Proceedings of the International Symposium on Experimental Robotics (ISER), 2012.

[22] H. Kress-Gazit, G. E. Fainekos, G. J. Pappas, "Translating Structured English to Robot Controllers," Acvanced Robotics 22, pp 1343-1359, 2008.

[23] J. Dzifcak, M. Scheutz, C. Baral, and P. Schermerhorn, "What to Do and How to Do It: Translating Natural Language Directives into Temporal and Dynamic Logic Representation for Goal Management and Action Execution," In IEEE International Conference on Robotics and Automation, May 2009.

[24] P. Lison, and G-J M. Kruijff, "An Integrated Approach to Robust Processing of Situated Spoken Dialogue," Proceedings of the Second International Workshop on the Semantic Representation of Spoken Language, pp 58-65, March 2009.

[25] R. Cantrell, M. Scheutz, P. Schermerhorn, and X. Wu, "Robust Spoken Instruction Understanding for HRI," Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction, pp. 275-282, Mar. 2010.

[26] M. Scheutz, R. Cantrell, and P. Schermerhorn, "Toward Humanlike Task-Based Dialogue Processing for Human Robot Interaction," AI Magazine, Vol. 32, No. 4, pp. 77-84, 2011.

[27] M. Scheutz, "Robust Natural Language Dialogues for Instruction Tasks," In Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 7692, http://dx.doi.org/10.1117/12.852179, 2010.