Facilitating Human-Robot Interaction: A Formal Logic For Task Description

Takehiko Nakama *, Enrique Muñoz, Kevin LeBlanc, and Enrique Ruspini

European Center for Soft Computing
c/ Gonzalo Gutiérrez Quirós, s/n, 33600 Mieres, Spain
{takehiko.nakama,enrique.munoz,kevin.leblanc,enrique.ruspini}@softcomputing.es
http://www.softcomputing.es/

Abstract. We develop a formal logic for task descriptions that are easy to interpret for both humans and robots. Tasks are described in propositional forms that reflect syntactic structures observed in natural language so that all the resulting task descriptions can be easily understood by humans. At the same time, these propositional forms ensure that each task description is interpretable by robots. Infinitely many task descriptions can be created in this formal logic so that the formal logic can support complex human-robot interactions. We establish a hierarchy of propositions that enhances the expressive power, the interactivity, and the deductive apparatus of our formal logic. We also examine how to systematically evaluate the feasibility of each task description using the formal logic.

Keywords: formal logic, propositional logic, predicate logic, quantificational logic, fuzzy logic, human-robot interaction, peer-to-peer communication, task description

1 Introduction and Summary

By interacting with humans, robots can effectively perform a wide range of practical tasks—for instance, assistance to people with disabilities (e.g., [25], [32], [12], [24]), search and rescue (e.g., [21], [7], [8], [30]), and space exploration (e.g., [36], [15], [14], [13]). The mode and degree of human-robot interaction vary considerably (see [18] for review). In the teleoperation mode of human-robot communication, robot operations require continuous low-level inputs from humans; consequently, even slight lapses in communications can degrade performance substantially, and the workload of the human operator can be undesirably high (e.g., [3], [6]). At the other extreme, autonomous robots that operate with no human input often perform complex tasks poorly compared to robots that are capable of collaborating with humans interactively (e.g., [1], [11], [5]). To overcome the disadvantages of the fully teleoperational and autonomous approaches, many studies have been conducted recently to develop robotic systems in which robots and humans collaborate as true team members through peer-to-peer human-robot interactions (e.g., [28], [10], [9], [19], [20]). One of the major challenges of this approach

* Research for the authors supported by the Spanish Ministry of Economy and Competitiveness through the project TIN2011-29824-C02-02 (ABSYNTHE)

is the increased complexity of the human-robot interactions in the peer-to-peer mode (e.g., [18]).

Various schemes have been developed for the peer-to-peer mode of human-robot interaction. Many studies have proposed schemes based on natural language (e.g., [27], [2], [26]). Although natural language is clearly an effective medium for human communications, it presents several major problems when used for human-robot communications. Natural language descriptions tend to be notoriously underspecified, diverse, vague, or ambiguous, so they often lead to error spirals that are hard to overcome (e.g., [37], [35], [33], [16], [17]). Restricting natural language to a small finite set of linguistic expressions mitigates these problems but does not fully support the extensive human-robot interactions required for complex tasks. Highly mathematical or symbolic expressions may be easy for experts (e.g., mathematicians, logicians, programmers, and system administrators) to understand, but naive users cannot be expected to be able to understand them; consequently, they are unsuitable for the variety of situations in which robotic systems need to interact with naive users as well as experts. Low-level sensory and motor signals and executable code are easy for robots to interpret, but they are cumbersome for humans and thus cannot, on their own, create an effective human-robot interface.

In this paper, we take a first step toward establishing a formal logic to effectively mediate the peer-to-peer mode of human-robot interaction. Our formal logic serves as a middle ground between the natural-language-based mode of human communication and the low-level mode of robotic communication. In this scheme, each task description is a formal proposition, called a propositional task description, and it can be easily understood by humans because it resembles the corresponding natural-language sentence. At the same time, we ensure that each propositional task description can be unambiguously interpreted and executed by robots. Our formal logic has high expressive power; it can create infinitely many propositional task descriptions, just as a natural language can create infinitely many sentences. Thus, the formal logic can support highly complex human-robot interactions required by peer-to-peer human-robot collaborations. As in other formal logics, we can infer and reason in our formal logic. We also establish a hierarchy of propositional task descriptions that enhances the expressive power, the interactivity, and the deductive apparatus of our formal logic.

This study is closely related to two lines of research: situation calculus (and other logical formalisms such as event calculus and fluent calculus) and precisiation language (or precisiated natural language). Situation calculus (e.g., [29], [31]) is a formal logic for representing actions, situations, and fluents and for reasoning about them (see also [23] for event calculus and [34] for fluent calculus). While it has contributed significantly to formalizing planning and programming in robotics, its logical expressions are rather technical and thus fail our requirement of comprehensibility for naive users. In contrast, we focus on developing a descriptive and deductive apparatus to establish a hierarchy of task descriptions that results in high expressive power and interactivity for robots and a variety of users, including those who have no expert knowledge on robotic systems. For this purpose, our formal logic employs propositional forms that reflect syntactic structures observed in natural language. In this regard, our major departure from situation calculus brings us closer to precisiation language, which is an integral part of Zadeh's computational theory of perceptions (e.g., [40], [38], [39]). In this theory, Zadeh introduced the concept of precisiated natural language (PNL), which refers to a set of natural-language propositions that can be treated as objects of computation and deduction. The propositions in PNL are assumed to describe human perceptions, and they allow artificial intelligence to operate on and reason with perception-based information, which is intrinsically imprecise, uncertain, or vague. Each proposition in PNL is translated into a precisiation language, which then expresses it as a set or a sequence of computational objects that can be effectively processed by machines (e.g., [40], [38], [39]). Our study significantly extends this line of research in two regards. To our knowledge, our study is the first to formulate the architecture or structure of the precisiation language; we accomplish this formulation by establishing a formal logic as a precisiation language. Also, our study substantially generalizes Zadeh's precisiation language. Zadeh considered the primary function of natural language as describing human perceptions, and his PNL and precisiation language only deal with perceptual propositions ([40], [38], [39]). However, the importance of natural language is not limited to describing human perceptions. In describing tasks, we need not only perceptual propositions but also action-related propositions. Our formal logic can be considered a generalized precisiation language that can handle both perceptual and action-related propositions.

The remainder of this paper is organized as follows. In Section 2, we explain propositional task descriptions that constitute our formal logic. In Section 3, we present a recursive definition of the syntax of the formal logic. In Section 4, we describe inference and reasoning in our formal logic. In Section 5, we develop a hierarchy of task descriptions. In Section 6, we discuss the generality of our scheme. In Section 7, we examine the semantics of our formal logic, which is used to represent and determine the degree of feasibility of each propositional task description.

2 Propositional Task Descriptions

In this section, we develop propositional task descriptions that enable effective human-robot interactions. As described in Section 1, we describe each task in a propositional form observed in natural language, and the resulting task descriptions form a formal logic. Thus, all well-formed formulas of the formal logic are easy to interpret for a variety of users.

In Section 2.1, we describe the components of such propositional task descriptions. In Section 2.2, we describe how to form an atomic proposition. In Section 2.3, we describe how to form a compound proposition.

In these sections, we will provide examples of rather simple task descriptions and explain our scheme in an intuitive manner. However, our scheme can be easily applied to systems that require more intricate task descriptions. In Section 6, we will discuss how our scheme can be used to mediate human-robot interactions with various degrees of complexity.

2.1 Components of propositional task descriptions

To establish our propositional task descriptions, we first define sets whose elements constitute propositions. These sets will be called component sets. Consider the following component sets:

- S denotes the set of all agents that can perform a task. For instance, we can have

$$S = \{robot\ 1, robot\ 2, robot\ 3, user\}. \tag{1}$$

- V denotes the set of all verbs that characterize actions required by tasks. For instance, we can have

$$V = \{call, find, deliver, go, move, press, put\}\,. \tag{2}$$

- O denotes the set of all objects that may receive an action in V or compose an adverbial phrase [see (4)]. For example, we can have

$$\begin{aligned} O = \{&box, button, table, room\ 1, room\ 2, building, \\ &robot\ 1, robot\ 2, robot\ 3, user, null\}\,. \end{aligned} \tag{3}$$

  Here the three robots and the user in S are also in O because they can also receive an action in V. The element denoted by "null" will be called the null element. In Section 2.2, we will explain how it is used in forming a task description.
- A denotes the set of all adverbial phrases that can be included in task descriptions. For instance, we can have

$$\begin{aligned} A = \{&in\ \gamma, from\ \gamma, from\ \gamma_1\ to\ \gamma_2, to\ \gamma, null \\ &\mid \gamma, \gamma_1, \gamma_2 \in O\}\,. \end{aligned} \tag{4}$$

  Again, the null element is denoted by "null," and its use is explained in Section 2.2.
- C denotes the set of all connectives that can be used to combine multiple propositions in forming compound propositions (see Section 2.3 for compound propositions). For instance, we can have

$$C = \{and, if, or, then, until, whenever\}. \tag{5}$$

The elements in these component sets are combined in a specified manner (see Section 2.2) to form propositions (propositional task descriptions) in our formal logic.

2.2 Atomic Propositions

In our formal logic, an atomic proposition is defined to be a tuple that consists of elements in component sets. Each admissible tuple structure is specified in the form of a cartesian product of component sets. To develop task descriptions that are easy to interpret for a wide variety of users, including naive users, we employ tuple structures that reflect syntactic structures observed in natural language. For the component sets described in Section 2.1, we can define each atomic proposition to be an SVOA clause

(The S, V, O, and A in SVOA stand for subject, verb, object, and adverbial phrase, respectively) by setting the admissible tuple structure to $S \times V \times O \times A$. The SVOA structure is observed in many languages, including English, Russian, and Mandarin. Using the null element in O and A, we can also generate SVO, SVA, and SV clauses. See the following examples of atomic propositions resulting from the component sets (1)–(4).

- $\frac{robot\ 1}{S}\ \frac{move}{V}\ \frac{null}{O}\ \frac{null}{A}.$
  We will omit instances of the null element and express it more simply as
  $\frac{robot\ 1}{S}\ \frac{move}{V}.$
- $\frac{robot\ 2}{S}\ \frac{find}{V}\ \frac{ball}{O}.\ \left(\frac{robot\ 2}{S}\ \frac{find}{V}\ \frac{ball}{O}\ \frac{null}{A}.\right)$
- $\frac{robot\ 1}{S}\ \frac{deliver}{V}\ \frac{box}{O}\ \frac{from\ room\ 1\ to\ room\ 2}{A}.$

A task will be described atomic if it is represented by a single atomic proposition. For humans, these task descriptions are easy to specify and understand. Meanwhile, the structural and lexical constraints substantially limit the diversity and flexibility of everyday language, so we can ensure that the resulting task descriptions are unambiguously interpretable for robots. As will be explained in Section 5, we establish a hierarchy of task descriptions, and at the lowest level of the hierarchy, each atomic task description is directly associated with an indecomposable, self-contained executable code. Thus, these atomic propositions can be considered building blocks of all task descriptions, and they constitute each task description at higher levels of the hierarchy.

There are several ways to deal with the undesirable or nonsensical atomic propositions that can be formed in $S \times V \times O \times A$. (Note that in formal logics, there can be well-formed formulas that are self-contradictory.) We can remove all such propositions from the cartesian product to ensure that each resulting atomic proposition is acceptable. (In this case, we abuse the notation and let $S \times V \times O \times A$ denote the "cleaned" cartesian product.) Also, we can consider them as always false so that they will never be executed in practice (see Section 7).

2.3 Compound Propositions

A compound proposition consists of multiple atomic propositions combined by one or more connectives in the component set C. With the examples (1)–(5) in Section 2.1, we can form the following compound propositions:

- $\frac{\frac{robot\ 1}{S}\ \frac{put}{V}\ \frac{ball}{O}\ \frac{in\ box}{A}}{\text{atomic proposition}}\ \frac{if}{C}\ \frac{\frac{user}{S}\ \frac{call}{V}\ \frac{robot\ 1}{O}}{\text{atomic proposition}}.$
- $\frac{\frac{robot\ 2}{S}\ \frac{go}{V}\ \frac{to\ room\ 1}{A}}{\text{atomic proposition}}\ \frac{whenever}{C}\ \frac{\frac{user}{S}\ \frac{press}{V}\ \frac{button}{O}}{\text{atomic proposition}}.$

In formal logic, parentheses are used to indicate the scope of each connective. In our formal logic, they disambiguate the manner in which tasks are performed:

$$\bullet\ \underbrace{\frac{robot\ 1}{S}\ \frac{go}{V}\ \frac{to\ room\ 1}{A}}_{\text{atomic proposition}}\ \frac{if}{C}\ \left(\underbrace{\frac{user}{S}\ \frac{call}{V}\ \frac{robot\ 1}{O}}_{\text{atomic proposition}}\ \frac{or}{C}\ \underbrace{\frac{user}{S}\ \frac{press}{V}\ \frac{button}{O}}_{\text{atomic proposition}}\right). \tag{6}$$

Tasks described by compound propositions will be referred to as compound tasks. These compound task descriptions are still quite easy for humans to specify and understand, and the syntactic structures imposed on the clauses and the compositions ensure effective interpretation and execution by robots.

There are many tasks that we can describe using conditions and imperatives. For instance, consider the compound task (6). In this description, the clause "*user call robot 1*" or "*user press button*" represents a condition that must be checked in determining whether to send the robot to room 1, and the clause "*robot go to room 1*" is an imperative. Our task descriptions consist of atomic propositions described in Section 2.2, and each atomic proposition will be either a condition or an imperative. By definition, an atomic task consists of a single atomic proposition, and the proposition is always an imperative. The type of each atomic proposition in a compound task description is unambiguously determined by the logical connective that connects it and by the location of the atomic proposition relative to the connective. For instance, an atomic proposition that forms a subordinate clause immediately following the connective "if" is considered a condition, whereas an atomic proposition that forms a clause immediately preceding the connective is considered an imperative. If the proposition is a condition, the robotic system monitors the described condition. If it is an imperative, the system executes the described action provided that all the required conditions are satisfied.

3 Well-Formed Formulas of the Formal Logic

We examine how to define the syntax of our formal logic for task description. We will provide a recursive definition of the syntax using the framework discussed in Section 2, but it can easily be generalized; we will address this issue in Section 6.

Using the component sets S, V, O, A, and C described in Section 2.1, we can define the syntax of a propositional logic as follows:

1. Any $x \in S \times V \times O \times A$ is an atomic well-formed formula.
2. If α and β are well-formed formulas, then $\alpha\ c\ \beta$, where $c \in C$, is also a well-formed formula.
3. Nothing else is a well-formed formula.

This recursive definition allows us to generate infinitely many well-formed formulas, i.e., propositional task descriptions that are unambiguously interpretable for humans and robots. With sufficiently rich component sets, we can describe any task that must be performed by the robotic system, and, using the syntax, we can ensure that the robotic system does not operate on any ill-formed task descriptions (i.e., task descriptions that are not interpretable).

With more extensive component sets, a predicate logic will be more appropriate as a formal logic for task description. To establish a predicate logic, we can consider each element in V as a predicate representing a ternary relation defined on $S \times O \times A$. Obviously, quantifiers can be incorporated in this formal logic. See Section 6.

4 Inference and Reasoning in the Formal Logic

As in other formal logics, we can infer and reason in our formal logic by adding a deductive apparatus to it. For inference and reasoning, we treat the connectives in the component set C as logic primitives (see, for instance, [35]). Syntactically, a set of inference rules can be constructed, and axioms can also be established. (The hierarchy described in Section 5 represents non-logical, domain-dependent axioms.) Typical introduction- and elimination-rules in formal logics, such as modus ponens and modus tollens, can be easily incorporated in our formal logic. Consequently, we can form a sequent, which consists of a finite (possibly empty) set of well-formed formulas (the premises) and a single well-formed formula (the conclusion), and we can examine its provability (derivability) using proof theory; we can determine if a conclusion follows logically from a set of premises by examining whether there is a proof of that conclusion from just those premises in the formal logic.

As will be described in Section 7, we can employ fuzzy relations to establish the semantics of our formal logic. This semantics allows us to investigate the truth conditions and the semantic validity of each proposition or sequent. As in other formal logics, comparative truth tables can be used to determine semantic validity.

5 Hierarchy of Task Descriptions

Using the scheme described in Sections 2–3, we develop a hierarchy of task descriptions. The importance of the hierarchy is threefold; (i) It enhances the expressive power of our task description scheme by building up its vocabulary while ensuring the interpretability of each resulting proposition; (ii) It enhances the interactivity of our scheme by allowing human-robot interactions to take place at various levels of detail; (iii) It strengthens the deductive apparatus of our formal logic by establishing domain-dependent axioms that can be used for inference and reasoning.

To explain this hierarchy, we first explore it with simple, concrete examples that appeal to ordinary practices and intuitions, and then move on to formal characterizations. A complex task can be described rather concisely, i.e., it can be expressed by an atomic proposition. In many cases, naive users will prefer specifying a complex task using an atomic proposition as opposed to a more lengthy compound proposition. On the other hand, in order for a robotic system to actually execute a complex task, the task must be broken down into simpler subtasks, and the manner in which the subtasks are performed must be specified. Consequently, the atomic proposition describing a complex task can be reexpressed as a compound proposition that consists of atomic propositions describing the required subtasks. Some of the subtasks may have to be further decomposed in order to fully specify how to execute them. Expert users may want to establish and combine these subtasks carefully so that the robotic system can perform the task effectively and efficiently. This flexibility in the level of detail gives naive users an efficient, user-friendly interface with robots while giving expert users the power to customize tasks. Since our interaction scheme must mediate interactions between robots and a variety of users, it should allow interactions to take place at various levels of detail. We develop a hierarchy of task descriptions to achieve this objective. The hierarchy determines the appropriate level of detail for each human-robot interaction, i.e., it determines whether, from the point of view of a given user, a given task is "atomic" or "compound."

Consider the following task description:

$$\frac{robot}{S}\ \frac{bring}{V}\ \frac{box}{O}\ \frac{from\ room\ 1\ to\ room\ 2}{A}. \tag{7}$$

This atomic proposition can be reexpressed as a compound proposition that consists of three atomic propositions representing the subtasks that must be performed; the task (7) can be defined as

$$\begin{aligned} &\frac{\frac{robot}{S}\ \frac{find}{V}\ \frac{box}{O}\ \frac{in\ room\ 1}{A}}{\text{atomic proposition 1}} \\ \frac{then}{C}\ &\frac{\frac{robot}{S}\ \frac{collect}{V}\ \frac{box}{O}}{\text{atomic proposition 2}} \\ \frac{then}{C}\ &\frac{\frac{robot}{S}\ \frac{go}{V}\ \frac{to\ room\ 2}{A}}{\text{atomic proposition 3}}. \end{aligned} \tag{8}$$

Atomic propositions 1 and 2 in (8) can also be reexpressed as compound propositions that clarify how they are performed; atomic proposition 1 in (8) can be defined as

$$\frac{\frac{robot}{S}\ \frac{go}{V}\ \frac{to\ room\ 1}{A}}{\text{atomic proposition}}\ \frac{then}{C}\ \frac{\frac{robot}{S}\ \frac{search}{V}\ \frac{box}{O}}{\text{atomic proposition}}, \tag{9}$$

and atomic proposition 2 in (8) can be defined as

$$\frac{\frac{robot}{S}\ \frac{go}{V}\ \frac{to\ box}{A}}{\text{atomic proposition}}\ \frac{then}{C}\ \frac{\frac{robot}{S}\ \frac{grasp}{V}\ \frac{box}{O}}{\text{atomic proposition}}. \tag{10}$$

Therefore, using (9)–(10) and atomic proposition 3 in (8), we can reexpress (7) as

$$\begin{aligned} &\frac{\frac{robot}{S}\ \frac{go}{V}\ \frac{to\ room\ 1}{A}}{\text{atomic proposition}}\ \frac{then}{C}\ \frac{\frac{robot}{S}\ \frac{search}{V}\ \frac{box}{O}}{\text{atomic proposition}} \\ \frac{then}{C}\ &\frac{\frac{robot}{S}\ \frac{go}{V}\ \frac{to\ box}{A}}{\text{atomic proposition}}\ \frac{then}{C}\ \frac{\frac{robot}{S}\ \frac{grasp}{V}\ \frac{box}{O}}{\text{atomic proposition}} \\ \frac{then}{C}\ &\frac{\frac{robot}{S}\ \frac{go}{V}\ \frac{to\ room\ 2}{A}}{\text{atomic proposition}}. \end{aligned} \tag{11}$$

Figure 1 visualizes the resulting hierarchy, which consists of three levels (levels 0, 1, and 2). For simplicity, each atomic proposition is represented by its verb; for instance, the atomic proposition at the highest level (level 2), "Robot *bring* box from room 1 to room 2," is represented by "bring." The task expressed by the atomic proposition at level 2 is described in more detail at the intermediate level (level 1), where the atomic propositions that involve the verbs "find," "collect," and "go" describe the subtasks that constitute the task. These subtasks are described in more detail at the lowest level (level 0), where they are expressed by the atomic propositions that involve the verbs "go," "search," and "grasp."

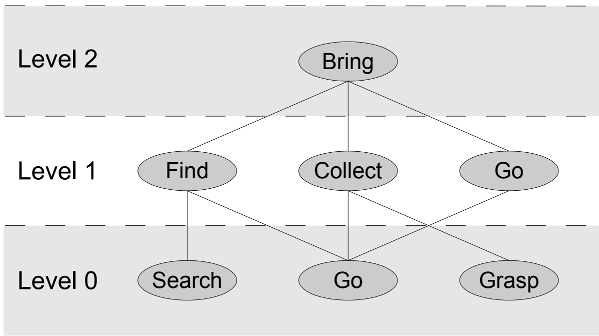

Fig. 1. Hierarchical task description. At the highest level (level 2), the task is expressed as atomic proposition (7), represented by "bring." At the intermediate level (level 1), the task is expressed as compound proposition (8), which consists of atomic propositions represented by "find," "collect," and "go." At the lowest level (level 0), the task is expressed as compound proposition (11), which consists of atomic propositions represented by "go," "search," and "grasp."

The suitability of a given task description depends on the level of granularity required for it. Naive users will most likely prefer describing tasks at level 2, thus preferring (7). For expert users, there may be situations where they 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 using (8) or at level 0 using (11) will be desirable. Thus, the hierarchy allows a variety of users to interact with robots at various levels of detail. At level 0, we have atomic propositions that are not decomposable; each of them is directly associated with a self-contained executable code that is run to perform the corresponding task.

The hierarchy of task descriptions can be characterized more formally as follows. Let S_i, V_i, O_i, A_i, and C_i denote the component sets for level i of the hierarchy ($i \geqslant 0$). The elements in these sets reflect the degree of detail suitable for level i. Then at level i, we establish a formal logic with these component sets, as described in Sections 2–3. Atomic propositions in $S_0 \times V_0 \times O_0 \times A_0$ are the building blocks of all propositional task descriptions; each of them is considered indecomposable, and it is directly associated with a self-contained executable code. For each $i \geqslant 1$, every atomic proposition in $S_i \times V_i \times O_i \times A_i$ can be decomposed into atomic propositions in $S_{i-1} \times V_{i-1} \times O_{i-1} \times A_{i-1}$. Figure 2 visualizes a typical form of this hierarchy. For each i and j, $p_j^{(i)}$ denotes an atomic proposition at level i, and e_j denotes an indecomposable, self-contained executable code associated with $p_j^{(0)}$. In the figure, $p_j^{(0)}$ is connected to e_j for each j, and for each $i \geqslant 1$ and j, $p_j^{(i)}$ is connected to atomic propositions at level $i-1$ that describe the subtasks associated with $p_j^{(i)}$. The figure shows that, for instance, $p_2^{(2)}$ can be expressed as a compound proposition consisting of two atomic propositions ($p_1^{(1)}$ and $p_4^{(1)}$) at level 1 and also as a compound proposition consisting of four atomic propositions ($p_1^{(0)}$, $p_2^{(0)}$, $p_3^{(0)}$, and $p_4^{(0)}$) at level 0.

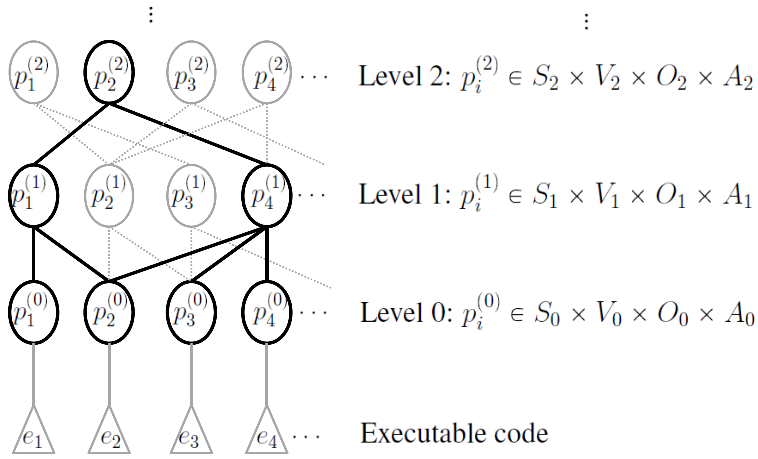

Fig. 2. Hierarchy of task descriptions. At level i, the formal logic described in Sections 2–3 is established with component sets S_i, V_i, O_i, A_i, and C_i. Atomic propositions in $S_0 \times V_0 \times O_0 \times A_0$ are the building blocks of all propositional task descriptions, and each of them is directly associated with an indecomposable, self-contained executable code. For each $i \geqslant 1$, every atomic proposition at level i is connected to atomic propositions at level $i-1$ that describe the subtasks associated with it.

Different levels of granularity may require different component sets, but the same syntactic structure is enforced at all levels. Using the hierarchy, we can ensure that all the resulting task descriptions remain interpretable for both humans and robots at each level, and we can achieve a high degree of efficiency in human-robot interaction by selecting the appropriate level of detail. As regards formal logic, the hierarchy represents non-logical, domain-dependent axioms that can be used for inference and reasoning, and these axioms can be used in examining the provability of a sequent in our formal logic (see Section 4).

6 Generality of Our Framework

Although we considered rather simple component sets and syntactic structures in Sections 2–5 to facilitate the exposition of our formal logic, we can easily extend our scheme to more sophisticated component sets and syntactic structures so that our formal logic can mediate human-robot interactions for complex robotic systems. Each component set can be made as large as necessary, and other component sets or clause structures can be incorporated in the formal logic. For instance, in addition to the SV, SVO, SVA, and SVOA structures described and used in Sections 2–5, we can also use other commonly observed clause structures (see, for instance, [4]), such as the SVC, SVOC, and SVOO structures, in forming atomic propositions. Furthermore, we can extend the clause structure so that a phrase can be used as the subject or the object in an atomic proposition. Negation, a unary logical connective, can certainly be included in C and incorporated in the formal logic.

As briefly mentioned in Section 3, we can establish not only a propositional logic but also a quantificational logic, which fully incorporates quantifiers and predicates in well-formed formulas. Since propositions that describe perceptions often include quantifiers (see, for instance, [38, 39]), a quantificational logic will be necessary to mediate extensive human-robot interactions regarding perceptions. Human-robot interactions are not limited to task descriptions, and our formal logic can mediate any form of human-robot interaction.

7 Semantics of the Formal Logic: Feasibility Evaluation

By establishing the formal logic for task descriptions, we can also develop a formal and efficient scheme for systematically evaluating the feasibility of each propositional task description. In fact, this scheme establishes the semantics of our formal logic. In formal logic, each proposition is evaluated and assigned a truth value—in two-valued logics, for instance, it is either 1 (true) or 0 (false). In our framework, we evaluate each proposition and let its truth value reflect the feasibility of the corresponding task specification; 1 indicates that the task certainly can be carried out whereas 0 indicates that it certainly cannot be. We maintain that it is more realistic and practical to let the degree of feasibility take on not only the values 0 and 1 but also other values between 0 and 1, so we use a many-valued logic in our formulation. In real-world problems, it can be highly practical to evaluate the feasibility of a task description before any serious attempt is made to execute it. In our case, for instance, we expect our robotic systems to interact with a variety of users, including users who have no knowledge of these systems, and some users may describe tasks that are virtually impossible to accomplish. It is more desirable to disregard such highly infeasible tasks immediately than to waste resources by attempting to realize them. Also, the user may want to be informed of the degree of feasibility of the task that he specifies before the system attempts to perform it. When multiple options are available for performing a specified task, the user may want to compare their degrees of feasibility before determining which option to take.

To evaluate the feasibility of each propositional task description systematically, we use a fuzzy relation, which is a generalization of a classical ("crisp") relation (see, for instance, [22]). It is a mapping from a Cartesian product to the set of real numbers between 0 and 1. While a classical relation only expresses the presence or absence of some form of association between the elements of factors in a Cartesian product, a fuzzy relation can express various degrees or strengths of association between them. (Hence a classical relation can be considered a "crisp" case of a fuzzy relation.) In our study, each propositional task description consists of pre-specified components, so a function that assigns a degree of feasibility to each proposition can be represented by a fuzzy relation on a Cartesian product of the components. Using some of the operations defined on fuzzy relations, we can systematically and economically determine the degree of feasibility of each task. We will provide details of the semantics in our full-length paper.

8 Discussions

We have taken a first step toward establishing a formal logic for task description that facilitates human-robot interaction. Various syntactic structures in natural language can be incorporated in our formal logic so that the resulting formal language serves as a middle ground between the natural-language-based mode of human communication and the low-level mode of machine communication. The high expressive power of the formal logic is achieved by its syntax, which enables us to create infinitely many task descriptions while ensuring that each of them is unambiguously interpretable. As in other formal logics, we can infer and reason in our formal logic. The hierarchy of propositions enhances the expressive power, the interactivity, and the deductive apparatus of our formal logic. Using the semantics of the formal logic, we can evaluate the feasibility of each task description systematically and efficiently.

We are currently implementing our task description scheme in a multi-agent robotic system. As described in Section 6, our formal logic is not limited to describing tasks. For instance, it can be applied to establishing perceptual propositions that allow artificial intelligence to operate on perceptual information obtained by humans. We are developing a quantificational logic that deals with such propositions.

References

1. K. A. Abbott, S. M. Slotte, and D. K. Stimson. Federal aviation administration human factors team report on: The interfaces between flightcrews and modern flight deck systems, June 1996.
2. J. F. Allen, D. K. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent. Toward conversational human-computer interaction. *AI Magazine*, 22:27–37, 2001.
3. M. Anderson, C. Conner, V. Daniel, M. McKay, and N. Yancey. Demonstration of the robotic gamma locating and isotopic identification device. In *Proceedings of the he American Nuclear Society Spectrum*, 2002.
4. Douglas Biber, Susan Conrad, and Geoffrey Leech. *A student grammar of spoken and written English.* Pearson ESL, 2002.
5. D.J. Bruemmer, D.A. Few, R.L. Boring, J.L. Marble, M.C. Walton, and C.W. Nielsen. Shared understanding for collaborative control. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(4):494–504, 2005.

6. D.J. Bruemmer, J.L. Marble, D.D. Dudenhoeffer, M. Anderson, and M.D. Mckay. Mixed-initiative control for remote characterization of hazardous environments. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 2003.
7. J. Casper and R. R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 33:367–385, 2003.
8. M. B. Dias, B. Kannan, B. Browning, E. G. Jones, B. Argall, M. F. Dias, M. Zinck, M. M. Veloso, and A. J. Stentz. Evaluation of human-robot interaction awareness in search and rescue. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2327–2332, 2008.
9. M. B. Dias, B. Kannan, B. Browning, E. G. Jones, B. Argall, M. F. Dias, M. Zinck, M. M. Veloso, and A. J. Stentz. Sliding autonomy for peer-to-peer human-robot teams. In *Proceedings of the 10th International Conference on Intelligent Autonomous Systems*, 2008.
10. M. Bernardine Dias, Thomas K. Harris, Brett Browning, Edward Gil Jones, Brenna Argall, Manuela M. Veloso, Anthony Stentz, and Alexander I. Rudnicky. Dynamically formed human-robot teams performing coordinated tasks. In *AAAI Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*, pages 30–38, 2006.
11. J. A. Espinosa, J. Cadiz, L. Rico-Gutierrez, R. E. Kraut, W. Scherlis, and G. Lautenbacher. Coming to the wrong decision quickly: Why awareness tools must be matched with appropriate tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 392–399. ACM, 2000.
12. D. Feil-Seifer and M. J. Mataric. Defining socially assistive robotics. In *Proceedings of the International Conference on Rehabilitation Robotics*, pages 465–468, 2008.
13. J. Ferketic, L. Goldblatt, E. Hodgson, S. Murray, R. Wichowski, A. Bradley, W. Chun, J. Evans, T. Fong, M. Goodrich, A. Steinfeld, and R. Stiles. Toward human-robot interface standards: Use of standardization and intelligent subsystems for advancing human-robotic competency in space exploration. In *Proceedings of the SAE 36th International Conference on Environmental Systems*, 2006.
14. T. Fong, I. Nourbakhsh, C. Kunz, L. Flückiger, J. Schreiner, R. Ambrose, R. Burridge, R. Simmons, L.M. Hiatt, and A. Schultz. The peer-to-peer human-robot interaction project. *Space*, 6750, 2005.
15. T. Fong and C. Thorpe. Vehicle teleoperation interfaces,. *Autonomous Robots*, 11:9–18, 2001.
16. M. Forsberg. Why is speech recognition difficult. *Chalmers University of Technology*, 2003.
17. P. Gieselmann and P. Stenneken. How to talk to robots: Evidence from user studies on human-robot communication. *How People Talk to Computers, Robots, and Other Artificial Communication Partners*, page 68, 2006.
18. M. A. Goodrich and A. C. Schultz. Human-robot interaction: a survey. *Found. Trends Hum.-Comput. Interact.*, 1:203–275, 2007.
19. M. Johnson, P. J. Feltovich, J. M. Bradshaw, and L. Bunch. Human-robot coordination through dynamic regulation. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2159–2164, 2008.
20. M. Johnson and K. Intlekofer. Coordinated operations in mixed teams of humans and robots. In *Proceedings of the IEEE International Conference on Distributed human-Machine Systems*, 2008.
21. H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahsahi, A. Shinjou, and S. Shimada. RoboCup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 739–743, 1999.
22. George J Klir and Tina A Folger. *Fuzzy sets, uncertainty, and information*. Prentice Hall, 1988.

23. R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):67–95, 1986.
24. V. Kulyukin, C. Gharpure, J. Nicholson, and G. Osborne. Robot-assisted wayfinding for the visually impaired in structured indoor environments. *Autonomous Robots*, 21:29–41, 2006.
25. G. Lacey and K. M. Dawson-Howe. The application of robotics to a mobility aid for the elderly blind. *Robotics and Autonomous Systems*, 23:245–252, 1998.
26. S. Lauria, G. Bugmann, T. Kyriacou, and E. Klein. Mobile robot programming using natural language. *Robotics and Autonomous Systems*, 38:171–181, 2002.
27. T. C. Lueth, T. Laengle, G. Herzog, E. Stopp, and U. Rembold. KANTRA: human-machine interaction for intelligent robots using natural language. In *Proceedings of the IEEE International Workshop on Robotand HumanCommunication*, pages 106–111, 1994.
28. J. Marble, D. Bruemmer, D. Few, and D. Dudenhoeffer. Evaluation of supervisory vs. peer-peer interaction with human-robot teams. In *Proceedings of the Hawaii International Conference on System Sciences*, 2004.
29. J. McCarthy. Situations, actions and causal laws. In M. Minsky, editor, *Semantic Information Processing*, pages 410–417. MIT press, Cambridge, MA, 1968.
30. I. R. Norbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion. Human-robot teaming for search and rescue. *Pervasive Computing*, pages 72–79, 2005.
31. R. Reiter. Proving properties of states in the situation calculus. *Artificial Intelligence*, 64:337–351, 1993.
32. I. Shim, J. Yoon, and M. Yoh. A human robot interactive system "RoJi". *International Journal of Control, Automation, and Systems*, 2:398–405, 2004.
33. B. Shneiderman. The limits of speech recognition. *Communications of the ACM*, 43(9):63–65, 2000.
34. Michael Thielscher. Introduction to the fluent calculus. *Electronic Transactions on Artificial Intelligence*, 3, 1998.
35. Paul Tomassi. *Logic*. Routledge, 1999.
36. B. Wilcox and T. Nguyen. Sojourner on mars and lessons learned for future planetary rovers. In *Proceedings of the SAE International Conference on Environmental Systems*, 1998.
37. T. Winograd and F. Flores. *Understanding computers and cognition: A new foundation for design*. Ablex Pub, 1986.
38. L. A. Zadeh. Some reflections on information granulation and its centrality in granular computing, computing with words, the computational theory of perceptions and precisiated natural language. *Studies in Fuzziness And Soft Computing*, 95:3–22, 2002.
39. L. A. Zadeh. Precisiated natural language (PNL). *AI Magazine*, 25(3):74–92, 2004.
40. Lotfi A Zadeh. A new direction in ai: Toward a computational theory of perceptions. *AI magazine*, 22(1):73, 2001.