Multihierarchical Interactive Task Planning: Application to Mobile Robotics

Cipriano Galindo, Juan-Antonio Fernández-Madrigal, and Javier González, *Associate Member, IEEE*

***Abstract*—To date, no solution has been proposed to human–machine interactive task planning that deals simultaneously with two important issues: 1) the capability of processing large amounts of information in planning (as it is needed in any real application) and 2) being efficient in human–machine communication (a proper set of symbols for human–machine interaction may not be suitable for efficient automatic planning and vice versa). In this paper, we formalize a symbolic model of the environment to solve these issues in a natural form through a human-inspired mechanism that structures knowledge in multiple hierarchies. Planning with a hierarchical model may be efficient even in cases where the lack of hierarchical information would make it intractable. However, in addition, our multihierarchical model is able to use the symbols that are most familiar to each human user for interaction, thus achieving efficiency in human–machine communication without compromising the task-planning performance. We formalize here a general interactive task-planning process which is then particularized to be applied to a mobile robotic application. The suitability of our approach has been demonstrated with examples and experiments.**

***Index Terms*—Hierarchical task planning, interactive task planning, mobile robots, world modeling.**

I. Introduction

THERE are a number of works in the literature that provide automatic planning processes with the capability of interacting with humans [49]–[52]. They are mostly based on sharing with the human a set of symbols that represent the environment, which is usually the same set used for the task-planning process. Sharing such a model with the human facilitates the subsequent sharing of intentions (as goals to achieve), processes (planning), and operating results (for guiding, acceptance, or correction of the plans). However, the reported symbolic representations and, particularly, the symbols needed for these applications may not be the most adequate for being understandable by every person or the best for planning efficiently (the real world can be categorized into different sets of symbols: some allowing for planning more efficiently than others and some with no semantics for a human).

Another important issue in the use of a symbolic representation for operating in the environment is the symbol grounding problem [2]–[4], which involves the maintenance of dynamic associations between symbols and their real-world counterparts. Out of the scope of interactive task planning, some works have addressed this, where the anchoring framework proposed in [1] is a relevant approach that copes with the symbol grounding problem by maintaining links between symbols and sensor data that refer to physical objects.

Finally, there is a third problem related to cognitive human–machine interaction that is rarely taken into account when a symbolic representation of the environment is used; most of the algorithms that process the information, particularly task planning, become very inefficient and even intractable when dealing with a realistic amount of data. We have found that this is an important problem in real environments [6] even when they are not large scale, for example, when they contain a nonnegligible number of objects that must be taken into account by the deliberative machine. This is particularly important in robotic applications.

In this paper, we focus on sharing cognitive information, particularly task planning and categorization of the physical world, between the human and a planning system in general. Although planning systems may belong to a variety of application fields or be executed on very different platforms, e.g., a personal computer, a workstation, or an embedded system, in this paper, we focus on the robotics arena and consider the planning system of a mobile robot. Thus, our approach should be seen as a particular case of human–machine interaction instead of human–robot interaction since we will relax the interfacing requirements, assuming that the components that permit the machine to communicate with humans (verbally, mechanically, visually, etc.) or to perform physically are already given.

In the last years, research on robotics has found several areas of interest apart from the classical search for predictable autonomous operation. Perhaps, one of the most promising is to consider the robot as a semiautonomous agent that is to be embedded into a human social environment, which involves considering the human–robot interaction as a nonnegligible aspect. A number of important issues have to be solved for constructing such robots: reliability, long-term operation, real-time performance, sharing of information and skills with humans, etc. There are a variety of them dealing explicitly with human–robot interaction since it can be studied ranging from the hardware–software interfaces, which permit the robot and the human to physically communicate with each other [35]–[38], to computational mechanisms for sharing cognitive information and processes [29], [40], emotions [30], [39], and intermediate levels such as assistance in basic skills like navigation in the case of assistant wheelchairs [8], [31], [34], manipulation [28], object delivering [43], [44], and many others [45]–[47].

Manuscript received September 25, 2007; revised December 11, 2007. This work was supported by the Spanish Government under Research Project CICYT-DPI2005-01391. This paper was recommended by Associate Editor J. Wang.

The authors are with the System and Engineering Department, University of Málaga, 29071 Málaga, Spain (e-mail: cipriano@ctima.uma.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2008.920227

1083-4419/$25.00 © 2008 IEEE

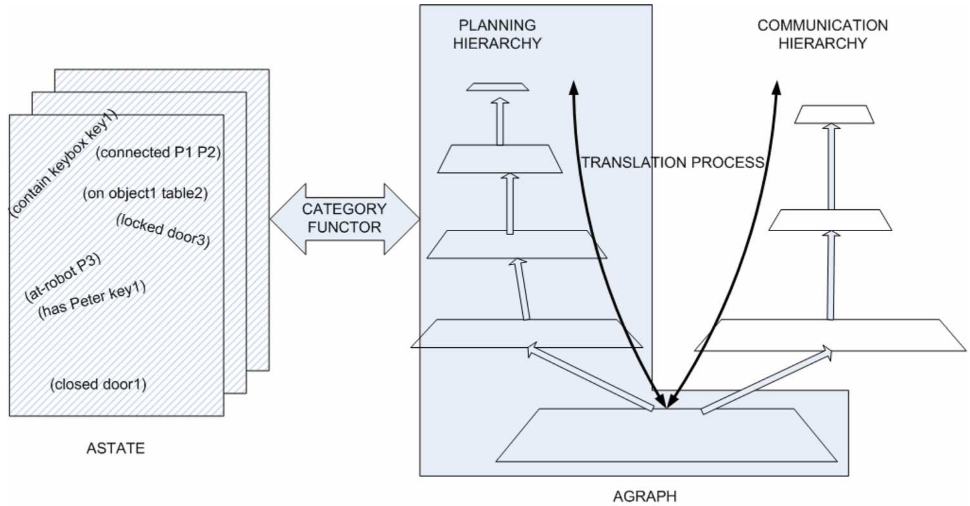

Fig. 1. General scheme of our interactive task planning approach based on multiple abstraction. In our multihierarchical model, (the shadowed one) the planning hierarchy is devoted to efficient planning, whereas the other is aimed to support symbolic human–machine communication. Our formalization also permits us to relate (left box) the classical planning to the symbolic abstraction.

Recently, some robotic architectures have been proposed [5], [32], [33] to provide the researchers with a suitable basis to address this diversity.

To address the interactive task planning issues previously commented, we propose an approach based on the inclusion into a symbolic graph model of the concept of abstraction [9], [19], [27], i.e., the structuring of symbols into hierarchies of detail, in such a way that task planning can use the different levels of abstraction as a heuristic to reduce computational cost and even make some intractable problems tractable [6], [10]. We have found that abstraction is also particularly relevant and suitable for addressing human–robot interaction at a cognitive level, and there has been psychological evidence that humans also use it, for example, to structure their cognitive maps[1] [4], [11]–[13].

However, we have also found [9] that the use of a single hierarchy of abstraction to reduce the complexity of a given operation has an aspect that is often neglected; a given hierarchy may be suitable to solve a specific task (for example, task planning) but not provide an effective solution to others (in our case, interacting with humans through understandable symbols). Thus, we explore here the use of more than one hierarchy in the same model, which is called multiple abstraction, which has previously demonstrated important improvements in tasks like graph search [9]. There is also psychological evidence on the use of multiple abstraction by human beings [17].

Our symbolic graph-based representation of the spatial environment includes multiple abstraction with the following characteristics: 1) it is suitable to use the anchoring paradigm (see [41]); 2) it is able to deal with the combinatorial explosion of classical planning processes (demonstrated previously in [6]); and 3) it is more efficient than using one symbolic representation both for planning and for human–machine interaction.

[1]The human cognitive map is the body of knowledge about the physical environment that is acquired and used, generally without concentrated effort, to find and follow routes from one place to another and to store and use the relative positions of places [18], [19].

We use one hierarchy to improve the efficiency of a classical task planning algorithm and a second one to communicate with the human efficiently in the sense that it contains the best symbols for human understanding (see Fig. 1 for an overall scheme of our approach). In the case that more than one person uses the application, several hierarchies for communication could be added. The planning hierarchy should be automatically constructed to improve the efficiency of planning (as has been reported previously in [7] and [25]). The communication hierarchy, on the other hand, is constructed under the preferences of a human.

The rest of this paper is structured as follows. Section II presents a formalization of our symbolic graph-based multihierarchical model of the environment where planning is to be done. Section III provides a formalization of human–robot task planning interaction in that context and an example of its possibilities. Section IV illustrates the use of our approach with a real experience. Finally, we outline some conclusions and future work.

II. Multihierarchical Symbolic Model

We have chosen an explicit symbolic representation of knowledge through annotated graphs [7], which allows us to manage any type of symbols and relations between them and also to include nonrelational information in the form of annotations. This representation has a direct utility, for example, in topological modeling of spatial environments, but here, we also use it to model objects (not only topological places) and relations different from spatial reachability. We have then enhanced the annotated graphs with a multihierarchical structure, obtaining the so-called multi-AH-graph, that is able to maintain several interconnected views of the machine workspace; some views are good for efficient planning, and others are good for human–machine communication.

In our previous works, the multihierarchical symbolic model, which was initially used only for robot path search in [7], has been exploited for different aims. In [6], a single-hierarchical model is used to perform efficient task planning, without the user participation into the process (the user only provides the goal of planning). The work presented in [25] copes with the automatic construction of the planning hierarchy of the model to minimize certain objective functions (like the cost of planning for the most common tasks), but, once again, without the user participation in the planning process. Finally, Galindo *et al.* [5] have reported a robotic control architecture featured with a single-hierarchical symbolic model of the space that permits a mobile robot to perform within large environments considering human participation in the plan execution but not during the planning process. The work that we present here copes with the user participation during the planning process. For that, we consider the general multihierarchical model presented in [7] but instantiated with two hierarchies; each one is devoted to solve a different problem (planning and user communication/interaction, respectively).

Section II-A presents a formalization of multi-AH-graphs using category theory [16]. This permits us to formally relate the graph-based representation of the environment, where plans are executed, to the clause-based representation of actions and states of the world needed by standard artificial intelligence (AI) planners, i.e., STRIPS, Metric-FF, etc. [14], [15]. Next, we propose an instantiation of our model with two hierarchies, namely, the planning hierarchy (Section II-B) to boost the planning process and the communication hierarchy (Section II-C) to fit the human communication needs and preferences.

A. *Formalization of Multihierarchical Graphs*

A multi-AH-graph is a graph-based symbolic representation of real environments that includes multihierarchical information, i.e., the possibility of abstracting groups of elements to superelements in different and simultaneous ways. From a constructivist point of view, we use a kind of abstraction that produces different layers (flat graphs) isolated from one another, which is called hierarchical levels; these represent the same environment with different amounts of detail. Vertexes of each hierarchical level represent elements of the world (e.g., places, objects, etc.) possibly annotated with nonrelational information (shapes, colors, etc.), whereas edges represent relations between them, possibly holding weights to indicate their strength. For example, vertexes can represent distinctive places for robot navigation, whereas a set of edges indicates the navigability relation between them with geometric distances as weights. The lowest hierarchical level is called the ground level; it represents the world with the maximum amount of detail available.

Formalization of Abstraction of Graphs: In order to propose a formalization of multi-AH-graphs, we first need to formalize abstraction of graphs.

Given two nonempty, finite, and directed multigraphs without loops G and H, each defined as a tuple (following [20])

$$(V, E, \gamma, \text{ini}, \text{ter})$$

where V is the finite set of vertexes, E is the finite set of edges, γ is the incidence function, ini is the initial function, and ter is the terminal function, an abstraction from graphs G to H can be defined as

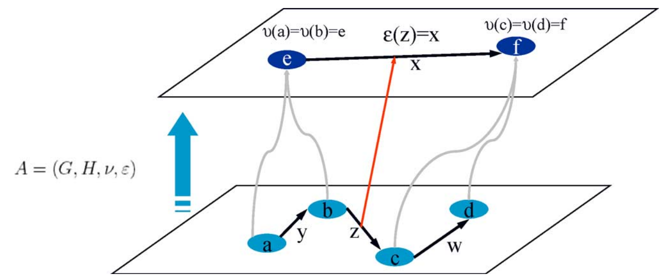

Fig. 2. Example of abstraction of graphs. Note that according to the restrictions imposed in (1), edges y and w cannot be abstracted.

$$A = (G, H, \nu, \varepsilon)$$

where G is the graph that is abstracted, H is the resulting graph, ν is the abstraction function for vertexes, and ε is the abstraction function for edges. The following restrictions must hold:[2]

$$\nu : V^G \rightarrow V^H \text{ is a partial function}$$
$$\varepsilon : E^G \rightarrow E^H \text{ is a partial function}$$

such that[3]

$$\begin{aligned}
&\forall z \in E^G, \text{def}(\varepsilon(z)) \Rightarrow \left[\text{def}\left(\nu\left(\text{ini}^G(z)\right)\right) \wedge \text{def}\left(\nu\left(\text{ter}^G(z)\right)\right)\right]\\
&\forall z \in E^G, \text{def}(\varepsilon(z)) \Rightarrow \left[\nu\left(\text{ini}^G(z)\right) \neq \nu\left(\text{ter}^G(z)\right)\right]\\
&\forall z \in E^G, \text{def}(\varepsilon(z)) \Rightarrow \begin{bmatrix}\nu(\text{ini}^G(z)) = \text{ini}^H(\varepsilon(z)) \wedge \\ \nu\left(\text{ter}^G(z)\right) = \text{ter}^H(\varepsilon(z))\end{bmatrix}. \qquad (1)
\end{aligned}$$

That is, an edge can be abstracted if and only if its initial and terminal vertexes have been abstracted into two different supervertexes (see Fig. 2). The vertex $\nu(a)$ for a given vertex $a \in V^{(G)}$ is called the supervertex of a. Analogously, the edge $\varepsilon(z)$ for a given edge $z \in E^{(G)}$ is called the superedge of z.

In the case that both υ and ε are total (every element of their domains has an image), the whole abstraction will be called complete. In the case that both υ and ε are on-to (every element of their ranges has a defined correspondence with an element in their domains), the whole abstraction will be called covered.

Functions ν and ε have inverses that are defined as

$$\nu^{-1} : V^H \rightarrow \text{power}(V^G)$$
$$\forall b \in V^H, \nu^{-1}(b) = \left\{a \in V^G : \nu(a) = b\right\}$$
$$\varepsilon^{-1} : E^H \rightarrow \text{power}(E^G)$$
$$\forall y \in E^H, \varepsilon^{-1}(y) = \left\{z \in E^G : \varepsilon(z) = y\right\}$$

where power(C) denotes the set of all the subsets of C. These inverse functions are called the refining functions for vertexes and edges, respectively. For any vertex $a \in V^H$, the vertexes

[2]In the following, we denote with a superscript the graph to which each component of the abstraction belongs. That is, V^G represents the set of vertexes of graph G.

[3]Given a function $f(x)$, def$(f(x))$ holds iif $f(x)$ is defined, i.e., there exists "y": $f(x) = y$.

belonging to $\nu^{-1}(a)$, if any, are called the subvertexes of a in G. Analogously, for any edge $z \in E^{(H)}$, the edges belonging to $\varepsilon^{-1}(z)$, if any, are called the subedges of z in G.

Formalization of Multi-AH-Graphs Through Category Theory: As it will be explained further on, multi-AH-graphs are mathematically finite subsets of the category of graphs with abstractions, which are called here as AGraph, which is similar to the well-known Category Graph of graphs with homomorphisms [21], except that it is defined under the aforementioned specification of graph abstraction, i.e., a partial morphism.

Thus, the AGraph category can be formally specified by

$$\mathrm{AGraph} = (\Theta, \nabla, \ell^-, \ell^+, I, \bullet)$$

where Θ is the collection of all possible nonempty, finite, and directed multigraphs without loops [20], ∇ is the collection of all possible abstractions of graphs following our definition of abstraction, ℓ^- is the lower hierarchical level function, ℓ^+ is the higher hierarchical level function, I is the identity function, and $\bullet$ is the composition of abstraction function, such that:

$\ell^- : \nabla \to \Theta$ and $\ell^+ : \nabla \to \Theta$ are functions that yield the two graphs involved in a given abstraction, i.e., for $A = (G, H, \nu, \varepsilon)$, $\ell^-(A) = G$, and $\ell^+(A) = H$.

$I : \Theta \to \nabla$, for any graph G, yields an abstraction that leaves it unaltered: $I(G) = (G, G, \nu_G, \varepsilon_G)$, where

$$\nu_G : V^G \to V^G \quad \forall a \in V^G, \nu_G(a) = a$$
$$\varepsilon_G : E^G \to E^G \quad \forall z \in E^G, \varepsilon_G(z) = z.$$

Finally, $\bullet : \nabla \times \nabla \to \nabla$ is a partial function that yields the composition of two given abstractions $A1$ and $A2$ as long as $\ell^+(A_1) = \ell^-(A_2)$ (otherwise, it is undefined). It is constructed as follows:

$$A_2 \bullet A_1 = \left(\ell^-(A_1), \ell^+(A_2), \nu_\circ, \varepsilon_\circ\right). \tag{2}$$

The two abstraction functions of $\bullet$ are defined by mathematical composition[4]: $\nu_\circ = \nu^{A_2} \circ \nu^{A_1}$ and $\varepsilon_\circ = \varepsilon^{A_2} \circ \varepsilon^{A_1}$.

This composition of abstractions is associative

$$\begin{aligned} &\forall G, H, J, K \in \Theta \\ &\qquad \forall A_1 = (G, H, \nu_1, \varepsilon_1) \\ &\qquad\ \ A_2 = (H, J, \nu_2, \varepsilon_2) \\ &\qquad\ \ A_3 = (J, K, \nu_3, \varepsilon_3) \in \nabla \\ &(A_3 \bullet A_2) \bullet A_1 = A_3 \bullet (A_2 \bullet A_1). \end{aligned} \tag{3}$$

Finally

$$\begin{aligned} &\forall G, H \in \Theta, \forall A = (G, H, \nu, \varepsilon) \in \nabla \\ &\qquad A \bullet I(G) = A = I(G) \bullet A. \end{aligned} \tag{4}$$

Under our definition for graph abstraction, constraints (2)–(4) can be easily demonstrated, and thus, AGraph is a category.

[4]The superscript denotes a particular element from a given abstraction, i.e., ν^{A_1} indicates the abstraction function for vertexes defined in the abstraction A_1.

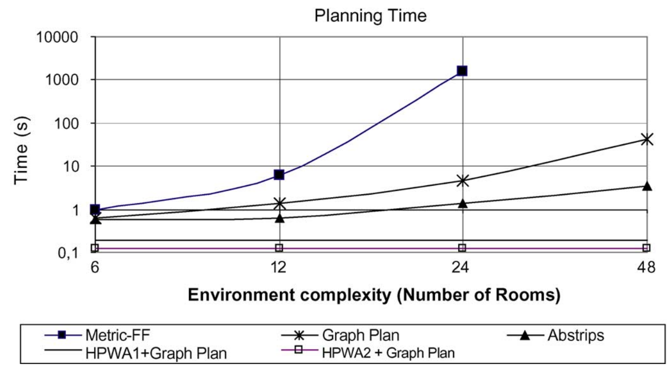

Fig. 3. Planning efficiency of HPWA. As long as the planning domain grows, the performance of classical planners largely decreases. The HPWA framework is capable of detecting and maintaining only the information that is necessary for solving the task at hand, ruling out irrelevant information and, thus, achieving high performance in complex and large environments.

For the purposes of this paper, we will consider a subcategory of AGraph that we will call CVAGraph* which stands for "Complete coVered AGraph with only connected graphs," i.e., a portion of AGraph with only complete and covered abstractions on graphs whose vertexes are all connected through some path. For practical interactive task planning, we will consider only multi-AH-graphs in which all the hierarchies (paths of abstraction) share a common ground hierarchical level. That level will serve as the link between the planning and the communication hierarchies.

B. Planning Hierarchy

Computational efficiency in classical task planners has been largely studied in the AI field but rarely in the robotics arena. However, in those applications in which a mobile robot performs within a real and large environment, i.e., an office building, classical AI planning often becomes an intractable problem [6]. For these cases, the most adopted solution relies on some heuristic mechanism to simplify the problem at hand. There are some approaches, which are known as hierarchical planners, that use some type of abstraction to speed up planning [22]–[24]. In this paper, we have employed hierarchical planning through world abstraction (HPWA) [6] which uses abstraction on the description of the world. The HPWA framework is basically a scheme that embeds an existing planner (the so-called embedded planner) to do the planning at different levels of a hierarchical representation of the environment, in order to improve planning efficiency in large and complex scenarios. First, a plan is found at a high level of abstraction of the world representation (low detailed). Then, that abstract plan is used to rule out irrelevant information at the next lower level (more detailed level) of the hierarchy, creating a sequence of plans more and more detailed. The performance of HPWA has been demonstrated elsewhere [6], comparing its efficiency to other classical planners when solving tasks in complex and large environments (see Fig. 3). In general, HPWA yields excellent results when planning in large environments, but its efficiency is largely tight to the particular hierarchical representation (symbolic hierarchy), and thus, a special attention should be paid to the automatic construction of appropriate hierarchies for planning. Although this topic is out of the scope of this paper, it deserves a brief explanation. Constructing the best hierarchy, i.e., the best arrangement of symbols for planning, is an intractable problem since it involves constructing all the possible hierarchies that can be constructed upon the set of ground data[5] and evaluating them with respect to the considered tasks. An additional problem is that the hierarchical representation should capture the dynamism of the environment with the possibility of modifying both the ground data (e.g., a new object has been found) and the tasks to be planned (e.g., the robot is commanded a new mission). Due to this complexity, there are few works in the literature addressing this topic, although it has been explored recently in [25], where an evolutionary algorithm is considered for continuously searching for a good hierarchy for planning under a set of variable robot tasks and also coping with environment dynamics.

AState Category and Its Relation to AGraph: In a nutshell, classical planning consists of searching an ordered sequence of actions, i.e., (GO origin destination), that modify the current state of the world (generally represented through sets of logical predicates) to attain a goal state. In our case, the symbolic world information required for planning is stored in our graph-based model, i.e., it is an annotated graph, and thus, a close relation between graphs and planning states must exist. Moreover, the HPWA works in a hierarchical fashion, and therefore, we must also establish a relation between graph abstraction and abstraction of planning states. All these relations can be formalized by defining a category for planning states with abstraction (called the AState Category) and relating it to the AGraph category through functors [16].

A planning state (a state for short) is a finite and consistent set of logical predicates in the form $p = (\text{predicate\_name}\ \text{param}_1, \ldots, \text{param}_k)$ that represents some piece of world information. We denote $\mho$ as the set of all possible planning states over a certain language[6] L, $\wp$ as the set of all possible symbol predicates, and Param as the set of all possible parameters that can be defined over L. Given a planning state $S \in \mho$, we will need a state parameter function $\text{SP}(S)$ that yields the distinct parameters from all logical predicates of S and a state predicate name function $\text{SN}(S)$ that yields the finite set of distinct predicate names of S. When an enumeration of elements from $\text{SP}(S)$ or $\text{SN}(S)$ is needed, we will use the notation $\text{SN}(S)[i](\text{SP}(S)[i])$ to refer to the ith element. We also extend the meaning of these functions to be applied to simple predicates rather than to states. That is, for the predicate, $p = (\text{at book-1 table-1})$, $\text{SN}(p) = \langle \text{at} \rangle$, $\text{SP}(p) = \{\langle \text{book-1} \rangle, \langle \text{table-1} \rangle\}$, and $\text{SP}(p)[2] = \langle \text{table-1} \rangle$.

Similar to the graph abstraction, we can define the abstraction of planning states. An abstraction A_{s} from a planning state S to a planning state T is a morphism between both states, which is defined as a tuple

$$A_{\text{s}} = (S, T, \xi, \pi)$$

[5]The number of the hierarchies that can be constructed upon a set of ground vertexes involves the Bell's number. For a very reduced environment, considering only ten vertexes, this figure is $B(10) = 115\,975$.

[6]In the rest of this paper, we assume that AState is defined on a certain first-order language L which will no longer be explicitly specified.

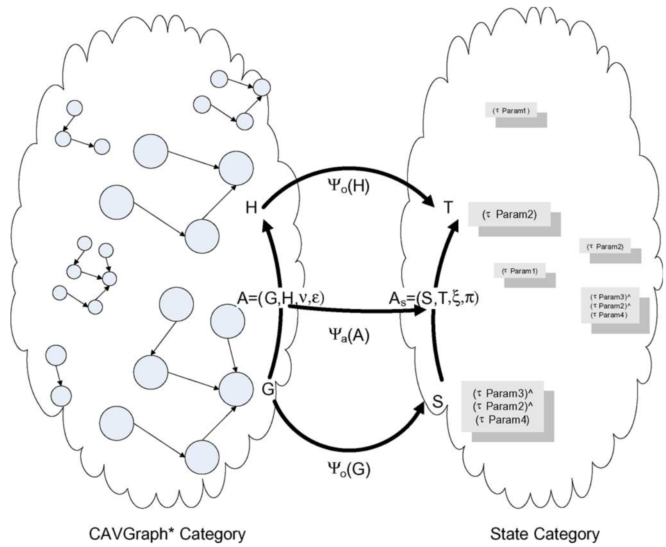

Fig. 4. Graph-state functor. Functor Ψ maps objects and arrows between the AGraph (the world) and AState (the planning space) categories. Its inverse is not always defined.

where

$$\xi : \text{SN}(S) \rightarrow \text{SN}(T) \text{ is a partial function}$$
$$\pi : \text{SP}(S) \rightarrow \text{SP}(T) \text{ is a partial function.}$$

Corresponding to the definition of AGraph, we formulate now the AState Category of planning states with abstractions. Given the first-order language L

$$\text{AState}(L) = (\mho, \blacktriangle, \varpi^-, \varpi^+, I, \bullet)$$

where $\mho$ is the collection of all possible states defined in L, $\blacktriangle$ is the collection of all possible abstractions on those states, ϖ^- is the refined state function, ϖ^+ is the abstracted state function, I is the identity function, and $\bullet$ is the composition of state abstraction function. Definitions are analogous to those given in Section II-A.

Once both categories are formalized, we can relate them by means of a functor. Functors are functions that relate objects and arrows of two categories, preserving their structures. We formalize the graph-state functor $\Psi = (\Psi_{\text{o}}, \Psi_{\text{a}})$ between the categories CVAGraph* and AState as follows:

$$\Psi_{\text{o}} : \Theta^* \rightarrow \mho$$
$$\Psi_{\text{a}} : \nabla \rightarrow \blacktriangle$$

where both of them are total functions. Informally, Ψ permits us to transform a graph into a planning state through Ψ_{o} and an abstraction of graphs into an abstraction of states that preserves the former transformation by means of Ψ_{a} (see Fig. 4). The complete formalization of Ψ requires the following auxiliary functions (the reader can safely skip these details and go to Section II-C).

The vertex-param translator Γ_ν is a partial, one-to-one, and on-to function from the set of vertexes of a graph to the set of all possible parameters considered in the planning domain (noted as Param). For instance, given a graph G, $\Gamma_\nu(a) = \langle\text{my desk}\rangle$, where $a \in V^{(G)}$, $\langle\text{my desk}\rangle \in \text{Param}$.

The edge-predicate translator Γ_e is a partial function from the set of edges of a graph to the set of all possible predicate names considered in the planning domain (noted as $\wp$). It is defined as follows:

$$\Gamma_e : E^\Theta \to \wp \times \wp \times \wp$$
$$\forall z \in E^\Theta, \Gamma_e(z) = (g, h,\ i) : g, h, i \in \wp.$$

For example, if a certain edge, namely, w, indicates a navigability relation between two places (vertexes), then $\Gamma_e(w)$ could be defined as $\Gamma_e(w) = (\langle\text{location}\rangle, \langle\text{location}\rangle, \langle\text{navigable}\rangle)$. That is, Γ_e transforms the relational information of edges (the world) into predicates (planning components) that refer to the same information. We also define three functions for retrieving separately each of the predicate names yielded by Γ_e. Thus

$$\forall z \in E^{(\Theta)}, \Gamma_e(z) = (g, h, i)$$
$$\Leftrightarrow \Gamma_{e1}(z) = g \wedge \Gamma_{e2}(z) = h \wedge \Gamma_{e3}(z) = i.$$

The edge-state translator β_G is then defined as a total function based on the definition of Γ_e and Γ_ν, that yields a set of logical predicates that represent the information represented by the edges in a given graph G and their related vertexes

$$\beta_G : E^G \to \mho$$
$$\forall z \in E^G, \beta_G(z) = \left\{ \begin{array}{c} p_1 = (\Gamma_{e1}(z) \quad \Gamma_\nu(\text{ini}(z))) \\ p_2 = (\Gamma_{e2}(z) \quad \Gamma_\nu(\text{ter}(z))) \\ p_3 = (\Gamma_{e3}(z)\ \ \Gamma_\nu(\text{ini}(z))\ \ \Gamma_\nu(\text{ter}(z))) \end{array} \right\}.$$

Informally, given a graph G, β_G transforms edges (the world) into three logical predicates (planning components) that express information regarding the vertexes involved in edges as well as about the type of relation between them. For instance, for the navigability edge w commented before that connects two vertexes a and b, such that $\Gamma_\nu(a) = \langle\text{my desk}\rangle$ and $\Gamma_\nu(a) = \langle\text{my office's door}\rangle$, the edge-state translator yields the following set of predicates:

$$\beta_G(w) = \{(\texttt{location}\langle\texttt{my desk}\rangle)$$
$$(\texttt{location}\langle\texttt{my office's door}\rangle)$$
$$(\texttt{navigable}\langle\texttt{my desk}\rangle\langle\texttt{my offices's door}\rangle)\}.$$

Finally, through the use of β_G, the Ψ_o functor can be directly defined as

$$\Psi_o : \Theta \to \mho$$
$$\forall G \in \Theta, \Psi_o(G) = \bigcup_{z \in E^G} \beta_G(z).$$

The second component of Ψ, the functor for arrows Ψ_a, permits us to transform abstractions of graphs into state abstractions, preserving the original structure. It can be defined as

$$\Psi_a : \nabla \to \blacktriangle$$
$$\forall A \in \nabla, \Psi_a(A) = A_s = \left(\Psi_o(G^A), \Psi_o(H^A), \tau, \kappa\right)$$

where functions τ and κ are defined as

$$\tau^{A_s} : \wp \to \wp$$
$$\forall z \in E^{(G^A)}, \begin{bmatrix} \tau^{A_s}(\Gamma_{e1}(z)) = \Gamma_{e1}(\varepsilon(z)) \\ \tau^{A_s}(\Gamma_{e2}(z)) = \Gamma_{e2}(\varepsilon(z)) \\ \tau^{A_s}(\Gamma_{e3}(z)) = \Gamma_{e3}(\varepsilon(z)) \end{bmatrix}$$
$$\kappa^{A_s} : \text{Param} \to \text{Param}$$
$$\forall a \in V^{G^A} : \kappa^{A_s}(\Gamma_\nu(a)) = \Gamma_\nu(\nu(a)).$$

Through functions τ and κ, the logical predicates that make up a planning state can be abstracted by following a given graph abstraction. That is, it is the medium by which the planning process can utilize the hierarchical information stored in the multi-AH-graph.

C. Communication Hierarchy

Psychology has revealed that humans seem to group symbols at different levels of detail in order to efficiently manage large amount of information [53]–[55]. For instance, when thinking about an office building, we rapidly sketch a hierarchical structure distinguishing groups of floors, sets of offices and corridors inside each floor, furniture at each office, etc. However, different individuals (even within the same environment) may exhibit particular preferences to group information and set different labels for identifying each group. For example, a cleaner may group a set of offices that share their dirtiness attaching them the label "hard work area," which is not done by a visitor. The same is obviously valid when different languages are used.

Within our model, the communication hierarchy is aimed to arrange world information in this humanlike manner. As will be explained further on, through this hierarchy, the user can interact with a robot using her/his own set of symbols and assigned labels. Since particular users may have different ways to model the same environment, her/his participation is needed in the construction of the hierarchy.

In this paper, we rely on an interactive construction process in which the user guides the robot to a particular place, i.e., the entrance door to an office, notifying that a symbol has to be created into the model with a certain label, i.e., "the door of my workplace." In this way, the user can provide the robot with a set of ground symbols, which are vertexes at the ground level of our multi-AH-graph and represent distinctive places or physical objects, and can also include relations between them, like connectivity or navigability, which are modeled through edges.[7] Subsequently, the user can select a set of vertexes to make up abstract symbols like rooms or areas. Fig. 5 shows an example of the communication hierarchy constructed in this way for a typical office environment.

[7]In the case of a robotic application, nonstructural data are automatically attached to vertexes and edges in the form of annotations, like robot pose, camera images, etc., for performing the operations that are planned (navigation, manipulation, etc.).

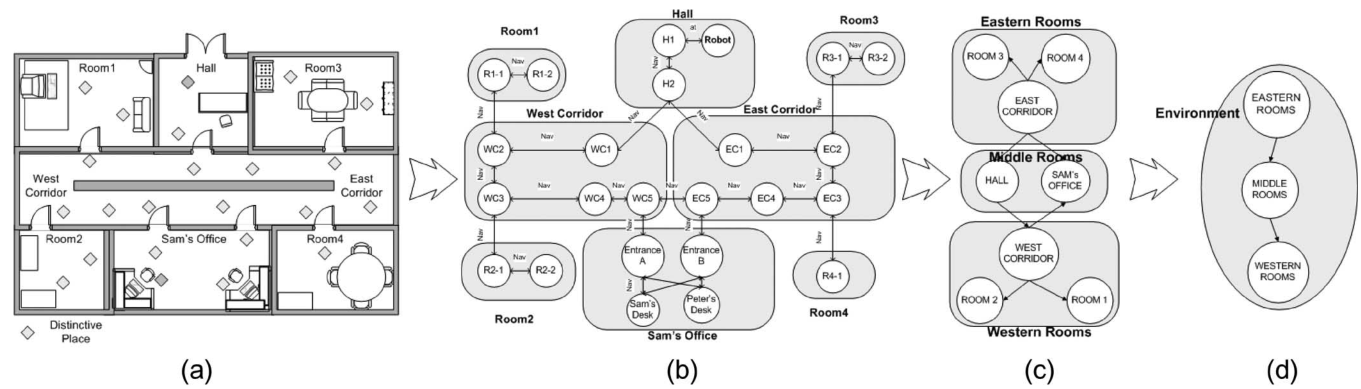

Fig. 5. Hierarchical representation of an office environment. (a) Environment. (b) Ground level: Topology of distinctive places. (b)–(d) Communication hierarchy for a particular user.

III. HUMAN–MACHINE INTERACTIVE TASK PLANNING

Our interactive task-planning process exploits the multihierarchical model presented in Section II. It permits the user to command a machine to solve a task through her/his own set of symbols (taken from the communication hierarchy), whereas the machine can efficiently plan the task using the planning hierarchy and communicate the results using concepts understandable by the user (via a translation of symbols from the planning hierarchy to the communication one). Moreover, when partial planning results are available, as in the HPWA case, the user can be reported with the evolution of the process, enabling her/him to guide the planning algorithm and propose modifications based on common-sense knowledge or particular preferences. Such an interactiveness can be deactivated by the user at any moment, making the hierarchical planner provide a final plan for the task at hand which is reported to the human by means of ground symbols.

Section III-A deals with the formalization of our interactive task-planning process. Section III-B illustrates all the modalities that our interactive process permits to the user.

A. Formalization of the Interactive Task-Planning Process

Our interactive task planning approach iteratively uses the following phases.

1) Translation of the user's requested goal, which is specified by using symbols of some level of the communication hierarchy, into the planning hierarchy. This is done by refining the symbols involved in the goal until reaching the common ground level of the multi-AH-graph. For example, if the user goal is "go to my workplace," the human symbol (my workplace) involved in the task is moved down to the ground level, choosing one of their subvertexes as a refined symbol, for example, my desk. Thus, the user goal turns into "go to my desk."
2) Abstracting the goal through the planning hierarchy for finding a first sketch of a plan at a high level of abstraction (low detail). Usually, this is done at the highest levels of the planning hierarchy for improving efficiency.
3) Reporting the abstract plan to the user and translating it to the communication hierarchy. The human decision about that plan (to reject or accept it totally or partially) may involve a backward translation to the planning hierarchy and a subsequent step of planning.

In stages 1) and 3), it is necessary to pass symbolic information from one hierarchy to the other, particularly plans. Informally, refining and abstracting plans within a hierarchy consists of refining/abstracting the sequence of actions involved in the plan. A plan action is the instantiation, with variables of the domain, of a plan operator that is composed of a pair of logical predicates $\langle$Precond, Postcond$\rangle$. If at a certain moment the logical predicates of Precond are satisfied, the action can be executed and the environment can be modified according to Postcond by adding or eliminating information.[8]

Abstracting a plan produces a more general one since it contains more abstract symbols. Conversely, refining a plan yields a set of more detailed plans covering all possible combinations of the refinements for the parameters of the plan. Both mechanisms can be formalized through the plan abstraction and the plan refinement functions defined as follows.

The plan abstraction function (PlanUp) serves to abstract plans that have been produced at a certain level of the planning hierarchy. For clarity, to define the PlanUp function, we first define the action abstraction function (ActionUp).

ActionUp is a partial function that abstracts actions of a plan, like (GO $r1$ $n1$ $n2$), if and only if their pre- and postconditions can be abstracted. Formally, given an abstraction in CVAGraph*, i.e., $A = (G, H, \nu, \varepsilon)$, and its corresponding abstraction in AState, i.e., $A_\mathrm{s} = (Q, S, \xi, \pi)$ (constructed by the Ψ_a functor), let $\Sigma_Q(\Sigma_S)$ be the set of all possible actions involving parameters only from the planning state $Q(S)$. The ActionUp function is defined for an action a as

$$\begin{aligned}
&\mathrm{ActionUp}_{A_\mathrm{s}} : \Sigma_Q \to \Sigma_S \text{ is a partial function}\\
&\forall a \in \Sigma_Q, \mathrm{def}\left(\mathrm{ActionUp}_{A_\mathrm{s}}(a)\right)\\
&\Leftrightarrow \forall p \in (a^{\mathrm{Precond}} \cup a^{\mathrm{Postcond}}), \mathrm{def}\left(\xi\left(\mathrm{SN}(p)\right)\right) \wedge\\
&\forall j \in 1..\left|\mathrm{SP}(p)\right|, \mathrm{def}\left(\pi\left(\mathrm{SP}(p)[j]\right)\right).
\end{aligned}$$

[8]Plan actions given in the form (action-name $\mathrm{param}_1, \ldots, \mathrm{param}_i$) can be considered as logical predicates. Similarly, complete plans, as sequence of actions, can be considered planning states and, thus, objects of the AState category.

When defined

$$\mathrm{ActionUp}_{A_\mathrm{s}}(a) = a' \in \Sigma_S : \left[\begin{array}{c} |\mathrm{SP}(a')| = |\mathrm{SP}(a)| \wedge \\ \mathrm{SN}(a') = \mathrm{SN}(a) \wedge \\ \forall j \in 1..\,|\mathrm{SP}(a')|\,, \mathrm{SP}(a')[j] = \pi\left(\mathrm{SP}(a)[j]\right) \end{array}\right].$$

The PlanUp can then be formalized. Given an abstraction in CVAGraph*, for example, $A = (G, H, \nu, \varepsilon)$, and its corresponding abstraction in AState, i.e., $A_\mathrm{s} = (Q, S, \xi, \pi)$, let $\mho_Q(\mho_S)$ be the set of all possible plans whose actions are in $\Sigma_Q(\Sigma_S)$. PlanUp is defined as

$$\begin{aligned} &\mathrm{PlanUp}_{A_\mathrm{s}} : \mho_Q \to \mho_S \\ &\forall p = (a_1, a_2, \ldots, a_n) \in \mho_Q \\ &\mathrm{PlanUp}_{A_\mathrm{s}}(p) = p' = (a'_1, a'_2, \ldots, a'_n) \in \mho_S : \\ &\forall a'_i \in p', a'_i = \mathrm{ActionUp}_{A_\mathrm{s}}(a_i). \end{aligned}$$

On the other hand, the plan refinement function (PlanDown) yields a refined version of a plan p based on refinement of graphs. As before, we first define the action refinement function (ActionDown) and then the equivalent definition for plans.

Given an abstraction in CVAGraph*, i.e., $A = (G, H, \nu, \varepsilon)$, and its corresponding abstraction in AState, i.e., $A_\mathrm{s} = (Q, S, \xi, \pi)$ (constructed by the Ψ_a functor), let $\Sigma_Q(\Sigma_S)$ be the set of all possible actions involving parameters from $Q(S)$. The ActionDown function is always defined,[9] and when it is applied to an action, a yields a set of actions $\{a'\}$ such that the conditions found near the bottom of the page are met.

Informally, through this definition, we establish that given an action a, it is refined on a set of actions that all have the same length and action name. In addition, we impose that all possible combinations of refined actions are considered and that several instances of a parameter in a must be refined to the same parameter in each refined action.

The PlanDown can now be formalized. Given an abstraction in CVAGraph*, i.e., $A = (G, H, \nu, \varepsilon)$, and its corresponding abstraction in AState, i.e., $A_\mathrm{s} = (Q, S, \xi, \pi)$, let $\mho_Q(\mho_S)$ be the set of all possible plans whose actions are in $\Sigma_Q(\Sigma_S)$. The PRF is defined as shown at the bottom of the page.

B. Sample Scenario for a Human–Robot Interactive Task Planning

Fig. 5 shows a scheme of part of a typical office environment. Upon the ground level, the communication hierarchy enables the robot to manage human symbols. On the other side, the planning hierarchy properly arranges the world elements with the goal of improving the task-planning process. Fig. 6 shows a multihierarchical representation of that environment.

In this scenario, let us consider the following application. An employee at the entrance of the office building is in charge of receiving and distributing mails to the rest of the employees. To facilitate his work, a servant robot can carry objects within the office building; thus, he has only to give the proper envelope to the robot and select the destination, i.e., "go to Sam's office."

As commented before, the first stage of the translation process consists of shifting the human concepts involved in the

[9]We assume in this paper that the refinement of actions is a total function; abstract symbols without subsymbols are not allowed (this is satisfied automatically when using CVAGraph*).

$$\begin{aligned} &\mathrm{ActionDown}_{A_\mathrm{s}} : \Sigma_S \to \mathrm{power}(\Sigma_Q) \\ &\forall a \in \Sigma_S \end{aligned}$$

$$\left[\begin{array}{c} \forall a' \in \mathrm{ActionDown}_{A_\mathrm{s}}(a) \\ \left[|\mathrm{SP}(a')| = |\mathrm{SP}(a)|\right] \wedge \left[\mathrm{SN}(a') = \mathrm{SN}(a)\right] \wedge \\ \left[\forall j \in 1..\,|\mathrm{SP}(a')|\,, \mathrm{SP}(a')[j] \in [\pi^{-1}]^{A_\mathrm{s}}\left(\mathrm{SP}(a)[j]\right)\right] \wedge \\ \left[(\exists x1, x2, \in 1..\,|\mathrm{SP}(a)| : \mathrm{SP}(a)[x1] = \mathrm{SP}(a)[x2]) \Rightarrow \mathrm{SP}(a')[x1] = \mathrm{SP}(a')[x2]\right] \wedge \\ \left[\bigcup_{a' \in \mathrm{ARF}_{A_\mathrm{s}}(a)} \mathrm{SP}(a') = \bigcup [\pi^{-1}]^{A_\mathrm{s}}\left(\mathrm{SP}(a)\right)\right] \wedge \\ \left[\forall j \in 1..\,|\mathrm{SP}(a)|\,, \forall b \in [\pi^{-1}]^{A_\mathrm{s}}\left(\mathrm{SP}(a)[j]\right), \exists a' \in \mathrm{ARF}_{A_\mathrm{s}}(a) : b \in \mathrm{SP}(a')\right] \end{array}\right]$$

$$\begin{aligned} &\mathrm{PlanDown}_{A_\mathrm{s}} : \mho_S \to \mathrm{power}(\mho_Q) \\ &\forall p, (a_1, a_2, \ldots, a_n) \in \mho_S,\ \mathrm{PRF}_{A_\mathrm{s}}(p) = \{p'\} \\ &\text{with } p' = (a'_1, a'_2, \ldots, a'_n) \in \mho_Q : \\ &\left[\forall a_i^s \in p, \forall a^* \in \mathrm{ActionDown}_{A_\mathrm{s}}(a_i^s), \exists p' \in \mathrm{PlanDown}_{A_\mathrm{s}}(p) : a_i^{s'} = a^*\right] \wedge \end{aligned}$$

$$\left[\begin{array}{c} \forall a'_\mathrm{u}, a'_\mathrm{w} \in p' \\ \left(\begin{array}{c} \exists x1 \in 1..\,|\mathrm{SP}(a'_\mathrm{u})| \wedge \exists x2 \in 1..\,|\mathrm{SP}(a'_\mathrm{w})| : \\ \pi^{A_\mathrm{s}}(\mathrm{SP}(a'_\mathrm{u})[x1]) = \pi^{A_\mathrm{s}}\left(\mathrm{SP}(a'_w)[x2]\right) \end{array}\right) \Rightarrow \mathrm{SP}(a'_\mathrm{u})[x1] = \mathrm{SP}(a'_\mathrm{w})[x2] \end{array}\right]$$

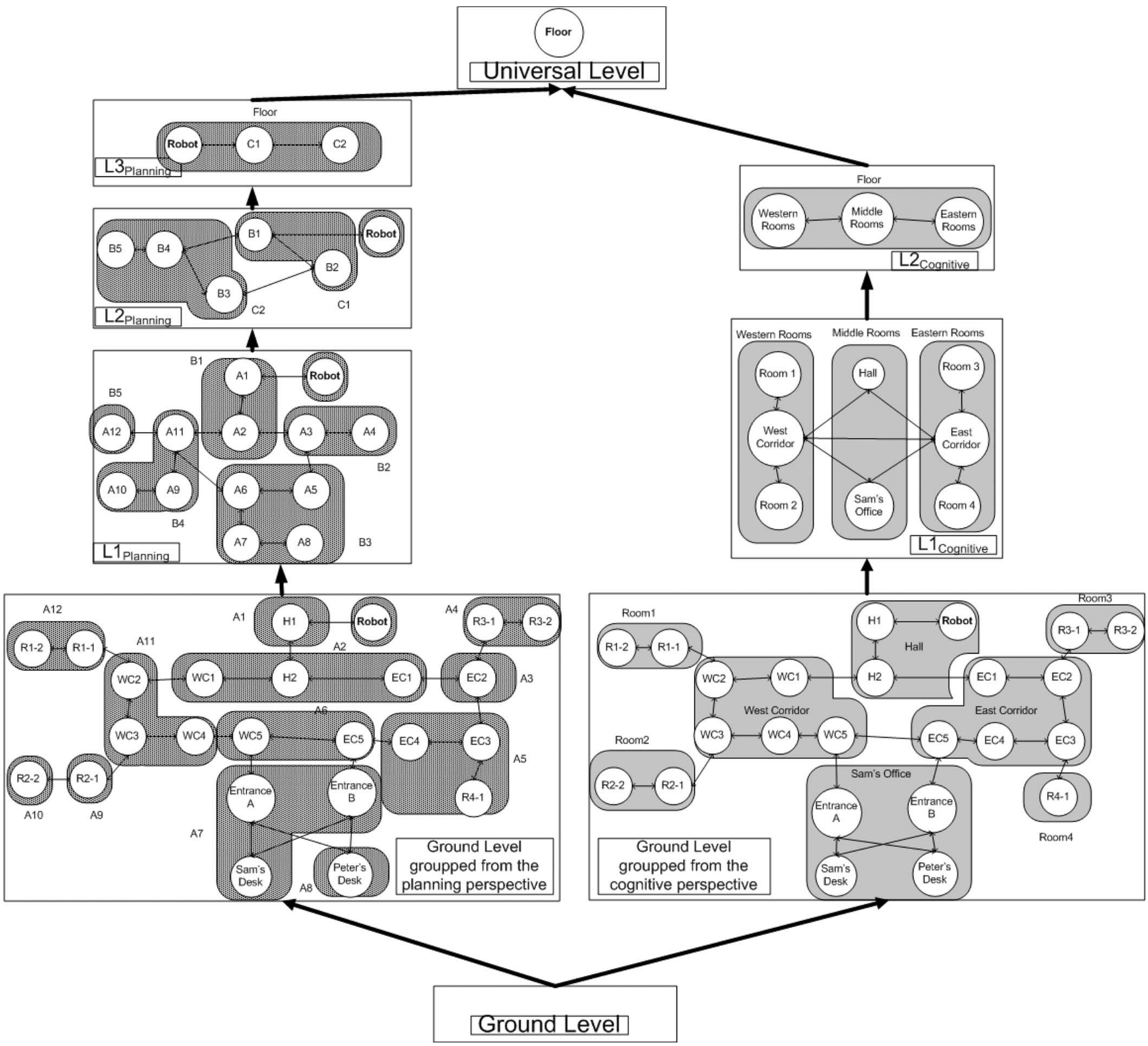

Fig. 6. Multihierarchical model of the environment shown in Fig. 4 with two hierarchies: (left branch) planning and (right branch) communication. Upon the ground level, shaded regions and labels indicate the grouping of vertexes into supervertexes. At the ground level, vertexes hold identifying labels that, for clarity sake, have been set to short codes.

requested goal into concepts of the ground level of the multi-AH-graph. This is solved by simply choosing any subsymbol (subvertex), for example, entrance to Sam's office. This selection may lead to the user's unsatisfaction (who probably would prefer Sam's desk to entrance to Sam's office as the destination) or to an unreacheable goal. Two possibilities are available then: choose another subvertex or ask the user for a more detailed specification of the goal.

Once an adequate specification of the goal is obtained at the ground level, the hierarchical planner solves the task using the planning hierarchy. HPWA first abstracts all the ground elements to the highest level of the planning hierarchy and then successively produces plans at different levels, which involve symbols that are not understandable by the user. Following our example, the planner produces the first abstract plan at level $\mathrm{L3}$ from the planning hierarchy[10] which is $(\mathrm{GO\ C1\ C2})$. This must be translated into human symbols; thus, it must be first refined to the ground level. The first refinement translates it to level $\mathrm{L2}$ of the planning hierarchy, yielding the list of plans

$$\mathrm{PlanDown}\left(\{(\mathrm{GO\ c1\ c2})\}\right)=\begin{Bmatrix}\{(\mathrm{GO\ b1\ b3})\}, \{(\mathrm{GO\ b1\ b4})\}\\ \{(\mathrm{GO\ b1\ b5})\}, \{(\mathrm{GO\ b2\ b3})\}\\ \{(\mathrm{GO\ b2\ b4})\}, \{(\mathrm{GO\ b2\ b5})\}\end{Bmatrix}$$

that can be simplified as a plan schema

$$\{(\mathrm{GO}\{\mathrm{b1}, \mathrm{b2}\}\ \{\mathrm{b3}, \mathrm{b4}, \mathrm{b5}\})\}.$$

[10]The abstract plan at the highest level is not considered in this example since it is trivial (there is a single node).

Level 1	{(GO {Hall, East-Corr., West-Corr., Room3} {Sam's Office, East and West Corr., Rooms 1,2,4})}
Level 2	{(GO {West, Middle, East} {West, Middle, East})}
Level 3	N/A

Fig. 7. Possible translations of the parameters of an abstract plan from L3 of the planning hierarchy.

By successively refining the abstract plans down through the planning hierarchy, a set of plans, which only involve parameters that represent distinctive places, is generated. Such plans are then abstracted up through the communication hierarchy via the PlanUp function. In this example, this yields the set of plans in the communication hierarchy shown in Fig. 7.

Through these translated plans, the user can proceed in the following ways.

1) Inquiring a more detailed plan. The translation of an abstract plan may not provide enough information to the user. In these cases, the user can request more information in two different ways: she can ask the robot for a translation of the same plan using more detailed symbols from the communication hierarchy (however, this increases the plan ambiguity), or she can ask the planning process for planning a new solution at a lower level of the planning hierarchy. In the latter case, the obtained plan will involve more detailed concepts which, in turn, can reduce ambiguity.
2) Rejecting part of a plan. Observe that even when the provided plan does not reveal enough information, the user can interact productively with the planning process, i.e., by rejecting certain spatial symbols. In our example, the user may require the robot to avoid the West-Corridor and Room2 regions for personal preferences or commonsense knowledge of the environment since they are, for example, crowded that day. Such discarded symbols are translated again into the planning hierarchy, reporting to the hierarchical planner that symbol $\mathrm{b4}$ at level $\mathrm{L2}$ must be discarded. Thus, HPWA plans now at level $\mathrm{L2}$ of the planning hierarchy without considering such a symbol, producing the new plan $\{(\mathrm{GO\ b1\ b2}), (\mathrm{GO\ b2\ b3})\}$, which is reported to the user as

$$\{(\mathrm{GO}\ \{\mathrm{Hall, East\text{-}Corridor}\}\ \{\mathrm{East\text{-}Corridor, Room3}\})\ (\mathrm{GO}\ \{\mathrm{East\text{-}Corridor, Room3}\}\ \{\mathrm{East\text{-}Corridor, Sam's\ Office}\})\}.$$

3) Suggesting an abstract plan. Due to the ambiguity involved in the translation process, the user may be informed about a set of different possibilities to solve a plan that may improve the efficiency of planning. She can select one out of the offered solutions based on her knowledge of the environment. For instance, in our example, the user can suggest the abstract plan {(GO Hall East-Corridor), (GO East-Corridor Sam's Office})} since she knows that it is not necessary to consider Room3 to arrive to the destination.

Thus, through the solution pointed out by the human, the planner can solve the task by considering only those symbols

Fig. 8. Robotic wheelchair SENA. It is based on a commercial powered wheelchair which has been endowed with several sensors and devices. The software architecture runs on the user's laptop, who can still use it for her/his work.

embraced by the ones suggested by her. In this example, the final plan at the ground level is {(GO h1 h2) (GO h2 ec1) (GO ec1 ec2) (GO ec2 ec3) (GO ec3 ec4) (GO ec4 ec5) (GO ec5 l2)} that would be communicated to the user with the labels given by her; see the following example:

{(GO "Hall's desk" to "Hall's door")
(GO "Hall's door" to "Hall connection to the East Corridor")
(GO "Hall connection to the East Corridor" to "Room3's door")
(GO "Room3's door" to "Corner of the East Corridor")
(GO "Corner of the East Corridor" to "Room4's door")
(GO "Room4's door" to "Sam's Office door B")
(GO "Sam's Office door B" to "Sam's Office Entrance B")}.

IV. Real Experiences

The effectiveness achieved by any system in which humans are involved is normally difficult to measure. This is the case of our human–computer interaction mechanism for which the user satisfaction should be somehow evaluated. The main concern of such an evaluation is that it largely depends on the user characteristics (i.e., cognitive abilities, age, gender, etc.), and thus, a representative sample of people should be selected, evaluating their opinion from psychological and cognitive points of view. This evaluation is out of the scope of this paper; thus, we rely on the study of a general case of interactive planning within a robotic application to show the suitability of our approach.

The robotic application that we consider here entails a real assistant robot (a robotic wheelchair) that provides mobility to physically impaired people within a large-scale building. The user is assumed to have enough cognitive abilities to take decisions and interact with the planning system of the

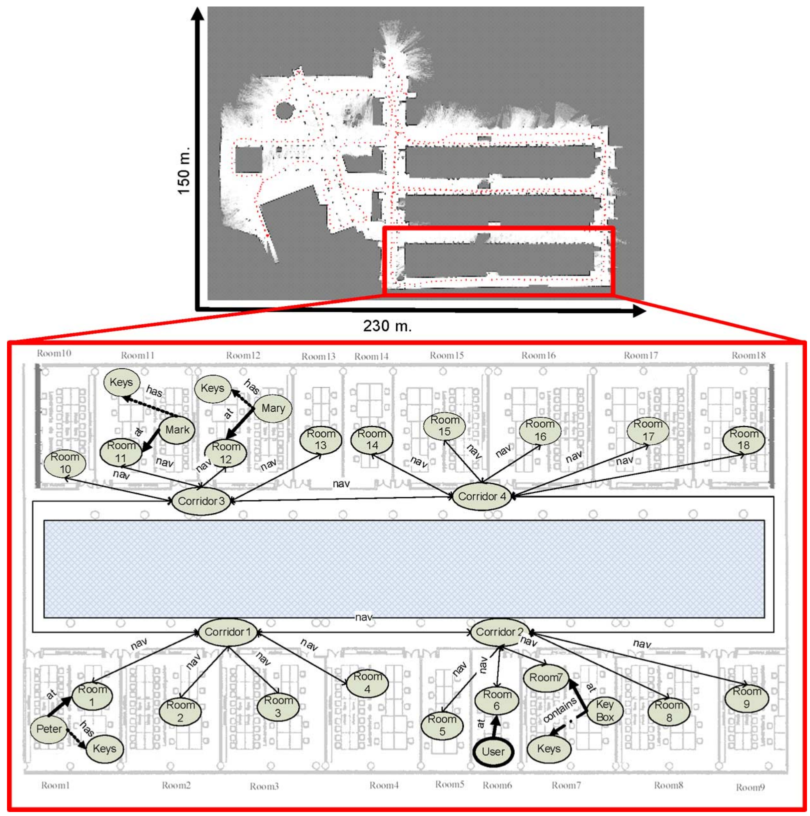

Fig. 9. Real scenario where the experiences of interactive task planning have been conducted. (Top) Whole environment entails four wings, a long corridor, and a patio. Small dots represent distinctive places. (Bottom) Part of the scenario where only the symbolic information of the first (not ground) level of the communication hierarchy is shown. Different line styles indicate different types of edges.

wheelchair as well as manual capabilities to open/unlock doors or manipulate objects. Our experiences have been conducted with the robotic wheelchair SENA (see Fig. 8), which includes a number of sensors for performing autonomous navigation and map building [26].

The successful performance of SENA is supported by the control architecture ACHRIN [5]. One of the main characteristics of this architecture is that it takes into account human abilities as part of the abilities of the robot, e.g., people can participate in the execution. This permits the system to ask the user for help when execution failures occur, as is the case of a locked door.[11] Through our interactive planning approach, the user can also participate during the planning phase, guiding the system to fulfill her/his requirements. User interaction with the planning system, and also with the rest of the robotic systems, mainly relies on verbal communication based on commercial voice recognition and text-to-speech tools. However, in dealing with large environments, voice interfaces may become hard to use due to its inability to properly recognize the user utterances, which is necessary to continuously repeat the commands. Thus, the user can also use graphical interfaces to control and interact with the robot.

[11] You can visit http://www.youtube.com/watch?v=D2oLBzlEEWA to watch one of our videos where SENA performs within an office scenario overcoming different situations like closed door, navigation errors, etc., owing to the user help. This video does not show the interactive planning but only the execution of a final plan.

In our experience, we do not consider replanning due to the execution failures (not solvable neither by the robot nor by the user). The planning system of ACHRIN is HPWA with FF metric as an embedded planner, and they have been implemented on C++ and improved with additional functions to cope with the interactive planning approach proposed in this paper.

Fig. 9 (top) shows a plan map of the considered scenario, which was constructed by a SLAM method [42], which is part of the Computer Science building at the University of Málaga. It is made of four wings connected through a long corridor and a patio. In our experiences, we focus on two of those wings with the particular setup shown in Fig. 9 (bottom, zoomed).

Room doors can be opened, closed, or locked. The actions that the user can perform are the following: open and unlock a door, search for a key in a key box, take a key, and ask anybody for a key, whereas the wheelchair only provides mobility (including obstacle avoidance, path and plan planning, etc.). Conventional (nonhierarchical) planning in this type of scenario with hundreds of distinctive places and objects becomes a complex, and even intractable, problem, as has been demonstrated in [6].

Fig. 9 (bottom) shows a scheme of the scenario with the vertexes (symbols) of the first level of the communication hierarchy involved in our experiences. We have considered an experience in which the wheelchair user is at his office (Room 6) and wants to go to Room 9 that is locked. Thus, the user commands the wheelchair at the first level of the communication hierarchy: "GO to Room 9," where its refinement is necessary

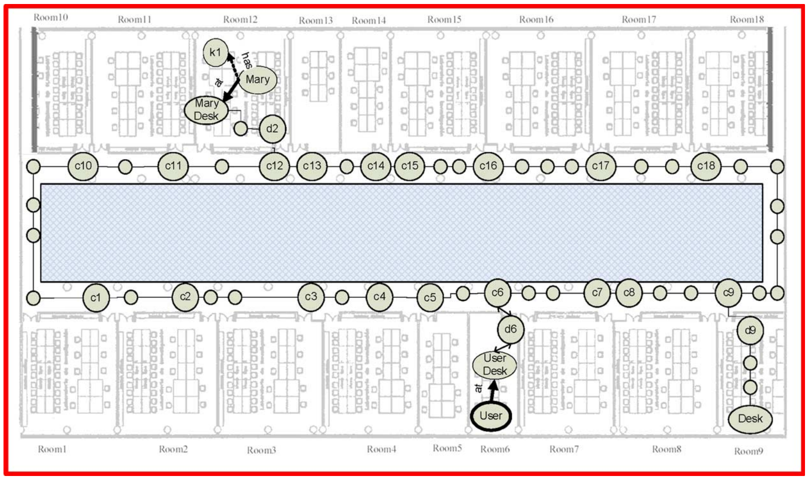

Fig. 10. Scenario where the involved ground symbols in the final plan accepted by the user are shown. For clarity in the picture, symbols that represent the entrance of offices have been rewritten as "cX," where "X" is the identifying office number; thus, for instance, "c9" represents the "entrance of office 9." Small circles represent intermediate distinctive places for navigation.

down to the ground level (partially shown in Fig. 10). An example of such a refinement would be "GO to Desk" (inside Room 9), which becomes the ground goal to be solved through the planning hierarchy.

The first solution to the proposed goal is communicated to the user, after the proper translation between hierarchies through the PlanDown and PlanUp functions, using symbols of the first level of the communication hierarchy (the same that the user employed in his initial command) as: "GO to Corridor 2," "GO to Room 7," "TAKE a Key from the KeyBox," "GO to Corridor 2," "GO to Room 9." Although this is the shortest and easier way to get to the destination, the user may prefer another solution which enables him to mix with his workmates. Thus, he asks the system for an alternative.

The next possible solution in this scenario is abstracted to the first level of the communication hierarchy through PlanUp as: "GO to Corridor 2," "GO to Corridor 1," "GO to Room 1," "ASK Peter for a Key," "GO to Corridor 1," "GO to Corridor 2," "GO to Room 9," that, although requires more execution time, would seem more acceptable for the user. Since the user knows that Peter is normally busy, he decides to phone him before going to his office in order to be sure that he has not left. Peter informs the user that he is going out to a meeting; thus, the user also rejects this plan since he knows that Peter will not be able to give him the key (this is something that the planning process was unable to foresee).

The planner then goes for another solution, yielding the plan: "GO to Corridor 2," "GO to Corridor 1," "GO to Corridor 3," "GO to $\{\text{Room } 11, \text{Room } 12\}$," "ASK $\{\text{Mark, Mary}\}$ for a Key," "GO to Corridor 4," "GO to Corridor 2," and "GO to Room 9." Note that there are four possibilities embedded in this plan: $(\text{Room } 11, \text{Room } 12) \times (\text{Mark, Mary})$. Any of these should be feasible (the user knows that both Mark and Mary are always at their offices), but the user still has certain preferences. For instance, he has strongly argued with Mark the same day; thus, he prefers not to meet him, choosing the second option: ask Mary for the keys.

Once this plan at the first level of the communication hierarchy is selected, the planning process refines it to the next lower level, which, in this case is shown at the bottom of the page.

In this situation, although the high-level plan of asking Mary for the key is accepted, the refined plan yielded by PlanDown involves passing close to Mark's office, which may also be rejected by the user who does not want to meet him at the corridor. Thus, he commands to reject the symbol "Door 11," and the alternative path for this task is planned and reported to the user as[12]: "GO out through door of office 6," "GO to entrance offices 7, 8, 9, 18, 17, 16, 15, 14, 13, 12," "GO in through door of office 12," "ASK Mary for key1," "GO out through door of office 12," "GO to entrance offices 13, 14, 15, 16, 17, 18, 9," "UNLOCK door of office 9," "GO in through door of office 9," "GO to Desk." This solution is, at last, accepted since it fulfills all the user's requirements and preferences. The "conversation" that has taken place between the user and the robot has been realized by using familiar concepts (symbols) of the former,

[12]When communicating plans, sequences of navigational tasks can be abbreviated, indicating the sequence of the destinations, as shown in this example.

"GO out through door of office 6," "GO to entrance office 5," "GO to entrance office 4," "GO to entrance office 3,"
"GO to entrance office 2," "GO to entrance office 1," "GO toentrance office 10," "GO to entrance office 11,"
"GO to entrance office 12," "GO in throughdoor of office 12," "ASK Mary for key 1," "GO out through door of office 12,"
"GO to entrance office 13," "GO to entrance office 14," "GO to entrance office 15," "GO to entrance office 16,"
"GO to entrance office 17," "GO to entrance office 18," "GO to entrance office 9," "UNLOCK door of office 9,"
"GO in through door of office 9," "GO to Desk"

which has produced an efficient interaction from the user's perspective. In addition, the planning process has produced plans efficiently in spite of being confronted with a large and complex scenario.

V. Conclusion

In this paper, we have proposed a symbolic model of the environment that makes the human–machine interactive task planning affordable in the real world. The use of multiple abstraction permits a semiautonomous agent to efficiently deal with large amount of information and with task planning. It also provides each user with the best set of symbols (those that the user understands well and that are according to his/her own experience) without compromising the efficiency. We have proposed a formalization of this multihierarchical model and of the task-planning process. Based on that, we have also presented a form of interactive task planning that permits a user and a robot to collaborate at very different moments of the planning process in order to improve the solutions. We have illustrated this with simulated and real experiences.

Further work will be devoted to introducing nonclassical planners into our approach (probabilistic methods) and using it for robots that are able to operate during long periods of time in crowded scenarios. We will also investigate automatic processes for learning the user preferences from his/her interaction with the system which could serve to regulate the amount of information reported to the user.

References

[1] S. Coradeshi and A. Saffiotti, "An introduction to the anchoring problem," *Robot. Auton. Syst.*, vol. 43, no. 2/3, pp. 85–96, May 2003.
[2] S. Harnand, "The symbol grounding problem," *Phys. D: Nonlinear Phenom.*, vol. 42, no. 1–3, pp. 335–346, Jun. 1990.
[3] C. S. Holding, "Further evidence for the hierarchical representation of spatial information," *J. Environ. Psychol.*, vol. 14, no. 2, pp. 137–147, 1994.
[4] S. C. Hirtle and J. Jonides, "Evidence of hierarchies in cognitive maps," *Mem. Cogn.*, vol. 13, no. 3, pp. 208–217, 1985.
[5] C. Galindo, J. A. Fernandez, and J. Gonzalez, "Control architecture for human–robot integration: Application to a robotic wheelchair," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 5, pp. 1053–1067, Oct. 2006.
[6] C. Galindo, J. Gonzalez, and J. A. Fernandez, "Hierarchical task planning through world abstraction," *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 667–690, 2004.
[7] J. A. Fernandez and J. Gonzalez, *Multi-Hierarchical Representation of Large-Scale Space*. Norwell, MA: Kluwer, 2001.
[8] J. A. Fernandez, C. Galindo, and J. Gonzalez, "Assistive navigation of a robotic wheelchair using a multihierarchical model of the environment," *Integr. Comput.-Aided Eng.*, vol. 11, no. 11, pp. 309–322, 2004.
[9] J. A. Fernandez and J. Gonzalez, "Multihierarchical graph search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 103–113, Jan. 2002.
[10] R. E. Korf, "Planning as search: A quantitative approach," *Artif. Intell.*, vol. 33, no. 1, pp. 65–88, 1987.
[11] B. Tversky, "Cognitive maps, cognitive collages, and spatial mental models," in *Proc. Int. Conf. COSIT*, 1993, vol. 716, pp. 14–24.
[12] E. H. Sandberg, "Constraints on the development of hierarchical spatial representations," *Cogn. Dev.*, vol. 14, pp. 597–619, 1999.
[13] H. Voicu, "Hierarchical cognitive maps," *Neural Netw.*, vol. 16, no. 5/6, pp. 569–576, Jun. 2003.
[14] A. L. Blum and M. L. Furst, "Fast planning through planning graph analysis," *Artif. Intell.*, vol. 90, no. 1, pp. 281–300, Feb. 1997.
[15] H. Jorg and N. Bernhard, "The FF planning system: Fast plan generation through heuristic search," *J. Artif. Intell. Res.*, vol. 14, pp. 253–302, 2001.
[16] D. E. Rydeheard and R. M. Burstall, *Computational Category Theory*. Hertfordshire, U.K.: Prentice-Hall, 1988.
[17] A. L. Patalano, E. E. Smith, J. Jonides, and R. A. Koeppe, "PET evidence for multiple strategies of categorization," *Cogn. Affect. Behav. Neurosci.*, vol. 1, no. 4, pp. 360–370, Dec. 2001.
[18] B. J. Kuipers, *The Cognitive Map: Could It Have Been Any Other Way? Spatial Orientation: Theory, Research, and Applications*, H. L. Picks and L. P. Acredolo, Eds. New York: Plenum, 1983.
[19] B. J. Kuipers, "The spatial semantic hierarchy," *Artif. Intell.*, vol. 119, no. 1/2, pp. 191–233, May 2000.
[20] M. R. Trudeau, *Introduction to Graph Theory*. New York: Dover, 1993.
[21] R. Kennaway, "Graph rewriting in some categories of partial morphisms," in *Proc. 4th Int. Workshop Graph Grammars Appl. Comput. Sci.*, H. Ehrig, H.-J. Kreowski, and G. Rozengerg, Eds., 1991, vol. 532, pp. 490–504.
[22] E. D. Sacerdoti, "Planning in a hierarchy of abstraction spaces," *Artif. Intell.*, vol. 5, no. 2, pp. 115–135, 1974.
[23] C. A. Knoblock, "Automatically generating abstractions for planning," *Artif. Intell.*, vol. 68, no. 2, pp. 243–302, 1994.
[24] J. Pineau, N. Roy, and S. Thrun, "A hierarchical approach to POMDP planning and execution," in *Proc. Workshop Hierarchy Memory Reinforcement Learn. (ICML)*, 2001.
[25] C. Galindo, J.-A. Fernández-Madrigal, J. González, A. Saffiotti, and P. Buschka, "Life-long optimization of the symbolic model of indoor environments for a mobile robot," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 5, pp. 1290–1304, Oct. 2007.
[26] J. Gonzalez, A. J. Muñoz, C. Galindo, J. A. Fernandez-Madrigal, and J. L. Blanco, "A description of the SENA robotic wheelchair," in *Proc. 13th IEEE Mediterranean Electrotech. Conf.*, Malaga, Spain, May 2006, pp. 437–440.
[27] F. Giunchiglia and T. Walsh, "A theory of abstraction," *Artif. Intell.*, vol. 57, no. 2/3, pp. 323–389, Oct. 1992.
[28] M. Ghorbel, M. Haariz, B. Grandjean, and M. Mokhtari, "Toward a generic human machine interface for assistive robots: The AMOR project," in *Proc. 9th ICORR*, 2005, pp. 168–172.
[29] A. Walsdorf and R. Onken, "Cognitive man–machine-cooperation: Modelling operators' general objectives and its role within a cockpit assistant system," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2000, pp. 906–913.
[30] R. R. Murphy, C. L. Lisetti, R. Tardif, L. Irish, and A. Gage, "Emotion-based control of cooperating heterogeneous mobile robots," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 744–757, Oct. 2002.
[31] R. Barea, L. Boquete, L. M. Bergasa, E. Lopez, and M. Mazo, "Electro-oculographic guidance of a wheelchair using eye movements codification," *Int. J. Robot. Res.*, vol. 22, no. 7/8, pp. 651–652, 2003.
[32] J. L. Fernandez, R. Simmons, R. Sanz, and A. R. Dieguez, "A robust stochastic supervision architecture for an indoor mobile robot," in *Proc. Int. Conf. Field Service Robot.*, 2001, pp. 269–274.
[33] T. W. Fong and C. Thorpe, "Vehicle teleoperation interfaces," *Auton. Robots*, vol. 11, no. 1, pp. 9–18, Jul. 2001.
[34] G. Bourhis, O. Horn, O. Habert, and A. Prusky, "An autonomous vehicle for people with motor disabilities," *IEEE Robot. Autom. Mag.*, vol. 8, no. 1, pp. 20–28, Mar. 2001.
[35] J. Scholtz, "Theory and evaluation of human–robot interaction," in *Proc. 36th Int. Conf. Syst. Sci.*, Big Island, HI, 2003.
[36] A. Nishikawa, T. Hosoi, K. Koara, D. Negoro, A. Hikita, S. Asano, H. Kakutani, F. Miyazaki, M. Sekimoto, M. Yasui, Y. Miyake, S. Takiguchi, and M. Monden, "Face MOUSe: A novel human–machine interface for controlling the position of a laparoscope," *IEEE Trans. Robot. Autom.*, vol. 19, no. 5, pp. 825–841, Oct. 2003.
[37] J. F. Borisoff, S. G. Mason, and G. E. Birch, "Brain interface research for asynchronous control applications," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 160–164, Jun. 2006.
[38] M. M. Moore, "Real-world applications for brain–computer interface technology," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 11, no. 2, pp. 162–165, Jun. 2003.
[39] W. D. Stiehl and C. Breazeal, "Affective touch for robotic companions—Affective interaction and systems and applications," in *Proc. 1st Int. Conf. ACII*, 2005, vol. 3784, pp. 747–754.
[40] S. S. Nudehi, R. Mukherjee, and M. Ghodoussi, "A shared-control approach to haptic interface design for minimally invasive telesurgical training," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 588–592, Jul. 2005.
[41] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. A. Fernandez-Madrigal, and J. Gonzalez, "Multi-hierarchical semantic maps for mobile robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, 2005, pp. 3492–3497.
[42] J. L. Blanco, J. A. Fernandez-Madrigal, and J. Gonzalez, "An entropy-based measurement of certainty in Rao-Blackwellized particle filter mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, Sep. 9–15, 2006, pp. 3550–3555.

[43] A. Agah and K. Tanie, “Human interaction with a service robot: Mobile-manipulator handing over and object to a human,” in *Proc. IEEE ICRA*, Albuquerque, NM, 1997, pp. 575–580.

[44] D. Kulic and A. Croft, “Strategies for safety in human robot interaction,” in *Proc. 11th Int. Conf. Adv. Robot.*, Coimbra, Portugal, 2003, pp. 644–649.

[45] K. W. Lee, H.-R. Kim, W. C. Yoon, Y.-S. Yoon, and D.-S. Kwon, “Designing a human–robot interaction framework for home service robot,” in *Proc. IEEE Workshop Robots Human Interactive Commun.*, 2005, pp. 286–293.

[46] S. Thrun, “Toward a framework for human–robot interaction,” *Hum.-Comput. Interact.*, vol. 19, no. 1/2, pp. 9–24, 2004.

[47] H.-R. Kim, K. Lee, and D.-S. Kwon, “Emotional interaction model for a service robot,” in *Proc. IEEE Workshop Robots Human Interactive Commun.*, 2005, pp. 672–678.

[48] F. Marrone, F. Raimondi, and M. Strobel, “Compliant interaction of a domestic service robot with a human and the environment,” in *Proc. 33rd Int. Symp. Robot.*, Stockholm, Sweden, 2002.

[49] Y. K. Hwang, P. C. Chen, and P. A. Watterberg, “Interactive task planning through natural language,” in *Proc. ICRA*, Minneapolis, MN, 1996, pp. 24–29.

[50] R. Dillmann, R. Zoellner, M. Ehrenmann, and O. Rogalla, “Interactive natural programming of robots: Introductory overview,” in *Proc. DREH*, Tolouse, France, 2002.

[51] H. Sun, B. Hujun, T. N. Man, and W. L. Fai, “Interactive task planning in virtual assembly,” in *Proc. ACM Symp. Virtual Reality Softw. Technol.*, London, U.K., 1999, pp. 174–175.

[52] S. Bagchi, G. Biswas, and K. Kawamura, “Interactive task planning under uncertainty and goal changes,” *Robot. Auton. Syst.*, vol. 18, no. 1, pp. 157–167, Jul. 1996.

[53] C. S. Holding, “Further evidence for the hierarchical representation of spatial information,” *J. Environ. Psychol.*, vol. 14, no. 2, pp. 137–147, Jun. 1994.

[54] S. C. Hirtle and J. Jonides, “Evidence of hierarchies in cognitive maps,” *Mem. Cogn.*, vol. 13, no. 3, pp. 208–217, 1985.

[55] T. P. McNamara, J. K. Hardy, and S. C. Hirtle, “Subjective hierarchies in spatial memory,” *J. Exp. Psychol.: Learn., Mem., Cogn.*, vol. 15, no. 2, pp. 211–227, 1989.

Cipriano Galindo was born in Málaga, Spain, in 1977. He received the M.S. and Ph.D. degrees in computer science from the University of Málaga, Málaga, in 2001 and 2006, respectively.

From 2004 to 2005, he was with the Applied Autonomous Sensor Systems, Örebro University, Örebro, Sweden. He is currently an Assistant Professor with the System and Engineering Department, University of Málaga. He is the author of almost 20 journal and conference papers and a book on symbolic world modeling and task planning for mobile robots. His research mainly focuses on mobile robotics, world modeling, and robot task planning and execution.

Juan-Antonio Fernández-Madrigal was born in Córdoba, Spain, in 1970. He received the M.S. and Ph.D. degrees in computer science from the University of Málaga, Málaga, Spain, in 1994 and 2000, respectively.

He is currently an Associate Professor (tenured) with the System and Engineering Department, University of Málaga. He is the author of almost 30 journal and conference papers and two books on multiple abstraction applied to mobile robotics. His main research interest during the last ten years has been the use of abstraction for intelligent robots.

Javier González (S’92–M’92–A’93) received the B.S. degree in electrical engineering from the University of Seville, Seville, Spain, in 1987, and the Ph.D. degree from the University of Málaga, Málaga, Spain, in 1993.

He was with the Department of “Ingenieria de Sistemas y Automatica,” University of Málaga in 1988. From 1990 to 1991, he was with the Field Robotics Center, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, working on mobile robots. He is currently an Associate Professor with the System and Engineering Department, University of Málaga, and he has been leading several Spanish and European projects on mobile robot since 1996. His research interests include mobile robot autonomous navigation, computer vision, and remote sensing. In these areas, he is the author and coauthor of more than 80 papers and three books about computer vision and mobile robot symbolic world modeling.